OLD DOMINION UNIVERSITY RESEARCH FOUNDATION

DEPARTMENT OF ELECTRICAL ENGINEERING
SCHOOL OF ENGINEERING
OLD DOMINION UNIVERSITY
NORFOLK, VIRGINIA

DESIGN OF MULTIVARIABLE FEEDBACK CONTROL SYSTEMS VIA SPECTRAL ASSIGNMENT

By

Roland R. Mielke, Principal Investigator

Leonard J. Tung, Co-Principal Investigator

and

Mohsen Marefat

Progress Report
For the period March 1, 1982 to September 30, 1982

Prepared for the
National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia 23665

Under
NASA Grant NSG-1650
Ruben Jones, Technical Monitor
Flight Dynamics and Control Division

(NASA-CR-169889) DESIGN OF MULTIVARIABLE FEEDBACK CONTROL SYSTEMS VIA SPECTRAL ASSIGNMENT Progress Report, 1 Mar. - 30 Sep. 1982 (Old Dominion Univ., Norfolk, Va.) 112 p HC A06/MF A01 CSCL 09B G3/63
N83-1833
Unclas
02789

January 1983

DEPARTMENT OF ELECTRICAL ENGINEERING
SCHOOL OF ENGINEERING
OLD DOMINION UNIVERSITY
NORFOLK, VIRGINIA

DESIGN OF MULTIVARIABLE FEEDBACK CONTROL SYSTEMS VIA SPECTRAL ASSIGNMENT

By

Roland R. Mielke, Principal Investigator

Leonard J. Tung, Co-Principal Investigator

and

Mohsen Marefat

Progress Report
For the period March 1, 1982 to September 30, 1982

Prepared for the
National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia 23665

Under
NASA Grant NSG-1650
Ruben Jones, Technical Monitor
Flight Dynamics and Control Division

Submitted by the
Old Dominion University Research Foundation
P.O. Box 6369
Norfolk, Virginia 23508-0369

January 1983

ABSTRACT

This report summarizes the progress of applied research conducted under NASA Grant NSG-1650 during the period March 1, 1982 to September 30, 1982. The objective of this project is to investigate the applicability of spectral assignment techniques to the design of multivariable feedback control systems. A fractional representation design procedure for unstable plants is presented and illustrated with an example. Then, a computer aided design software package implementing eigenvalue/eigenvector design procedures is described. A design example which illustrates the use of the program is explained.

PRECEDING PAGE BLANK NOT FILMED

TABLE OF CONTENTS

DESIGN OF MULTIVARIABLE FEEDBACK CONTROL SYSTEMS VIA SPECTRAL ASSIGNMENT

By

Roland R. Mielke[1], Leonard J. Tung[2] and Mohsen Marefat[3]

1. INTRODUCTION

This report summarizes the progress of applied research conducted under NASA Grant NSG-1650 for the period March 1, 1982 to September 30, 1982. The objective of this work is to investigate the applicability of spectral assignment techniques to the design of multivariable feedback control systems.

First, development of new frequency domain fractional representation design procedures for unstable plants is presented. The procedure consists of a technique for searching among all stabilizing controllers for those that also satisfy certain design specifications. Controller complexity and hidden system modes are considered. The procedure is illustrated with a design example. Then a new computer aided design software package implementing the time domain eigenvalue/eigenvector assignment procedures is described. The use of the program is illustrated with a design example. The program listing is included in the Appendix.

2. FRACTIONAL REPRESENTATION DESIGN PROCEDURES

2.1. Introduction

Our investigation in the area of frequency-domain controller design began with a study of the work by Youla and others (refs. 1,2). Among the many contributions in Youla's work is a procedure which leads to the

[1] Associate Professor, Department of Electrical Engineering, Old Dominion University, Norfolk, Virginia 23508.
[2] Assistant Professor, Department of Electrical Engineering, Old Dominion University, Norfolk, Virginia 23508.
[3] Graduate Research Assistant, Department of Electrical Engineering, Old Dominion University, Norfolk, Virginia 23508.

characterization of a general class of stabilizing compensators for a plant imbedded in a single-loop feedback control system. This procedure has then been generalized by Desoer and others (ref. 3) to form the basis of the so-called fractional representation approach. This approach offers a systematic procedure for constructing stabilizing compensators that achieve other design objectives such as decoupling the outputs and tracking step inputs. It should be noted that the objective of stabilization is resolved before other design objectives. In contrast to this type of approach is the work by Sain and others (refs. 4,5). Sain's work develops a direct method for the construction of compensators for a plant imbedded in a unity feedback control system. In this method, compensators that achieve design objectives such as decoupling are first constructed and then the issue of stabilization is resolved. Combining the results by Desoer and by Sain, we have developed design procedures that simultaneously achieve the design objectives of stabilization, decoupling, and tracking step inputs. These design procedures are expressed so that it is relatively easy to address the problems of complex compensators and unwanted hidden modes as noted in references 5-7.

In this report, we begin with a brief review of the fractional representation approach. After the review we outline two sets of procedures, one for stable plants and one for unstable plants, for constructing compensators that achieve design objectives of stabilization, decoupling, and tracking step inputs. These design procedures also allow us to construct simple compensators $C = -P^{-1} T(I-T)^{-1}$ for a given plant P by choosing simple stable diagonal T which satisfies certain requirements. The details of the procedures are exemplified by a problem of compensator design for an unstable plant. Finally, the problem of hidden modes is dealt with by carefully choosing the zeros of I-T.

2.2. Compensator Design

Consider the single feedback loop multivariable control system shown in figure 1. With the plant P(s) (a proper rational matrix) given, it is desired to design a controller C(s) (another proper rational matrix) for stabilization, decoupling and tracking step inputs. The fractional representation approach (refs. 3,8,9) offers a systematic procedure for achieving these design objectives. In this approach, the plant is expressed in a right and a left coprime exponential stable rational fractional descriptions, $P = N_r D_r^{-1} = D_\ell^{-1} N_\ell$ with U_r, V_r, U_ℓ and V_ℓ such that $U_r N_r + V_r D_r = N_\ell U_\ell + D_\ell V_\ell = I$. Note that all terms except possibly the plant P are proper rational matrices with poles in the open left-half complex plane. With these descriptions for P, a general class of stabilizing controllers is given by

$$C = (WN_\ell + V_r)^{-1}(-WD_\ell + U_r), \tag{1}$$

where W can be any proper exponential stable rational matrix as long as $WN_\ell + V_r$ is nonsingular. With this class of controllers, the closed-loop transfer function is given by

$$T = N_r[-WD_\ell + U_r]. \tag{2}$$

When P itself is exponential stable, we can have $N_r = N_\ell = P$, $D_r = D_\ell = I$, $U_r = U_\ell = 0$ and $V_r = V_\ell = I$. Equations (1) and (2) then become

$$C = - W[I+PW]^{-1} \tag{3}$$

and

$$T = - PW. \tag{4}$$

Equations (1) through (4) display the freedom in choosing a stabilizing controller as the freedom in choosing W. This freedom in choosing W can then be explored for achieving other design objectives such as decoupling and tracking. For decoupling, T is to be made diagonal. For tracking step inputs, every term in $I-T$ should have a zero at $s = 0$.

ORIGINAL PAGE IS
OF POOR QUALITY

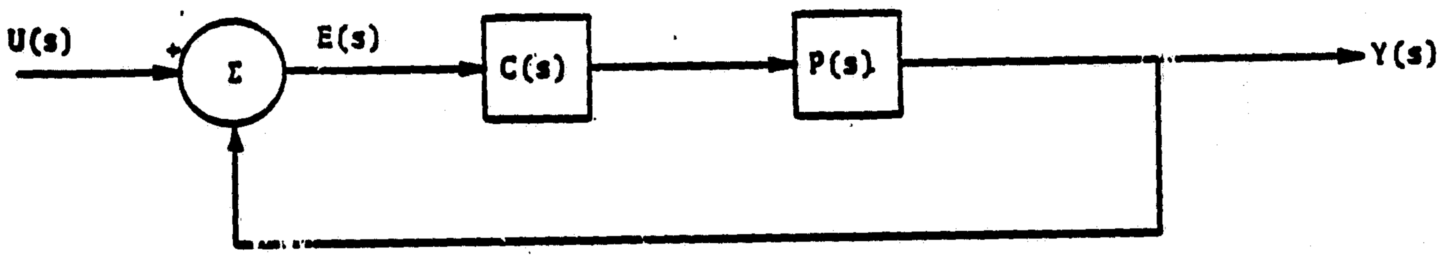


Figure 1. Multivariable feedback control systems.

Stable Plants

First assume that P is exponential stable and invertible. Under this assumption, $T = -PW$ is invertible if W is invertible. The invertibility of T is important because it eliminates the possibility of zero diagonal terms in T after T is made diagonal for decoupling, a case which indicates redundancy of certain input and output signals. For maintaining the stability of the closed-loop system, our approach is to choose stable T to make $W = -P^{-1}T$ stable. For decoupling, we only have to work with diagonal T. For tracking step inputs, we must choose among those matrices T such that all terms in I-T have a zero at $s = 0$. For constructing controllers that simultaneously achieve stabilization, decoupling and tracking step inputs, we thus have the following procedures:

(i) For decoupling, choose $T = \text{diag}\{T_1,\ldots,T_n\}$.

(ii) Let $P^{-1} = [\tilde{P}_1,\ldots,\tilde{P}_n]$, where $\tilde{P}_i$ is the ith column of P^{-1}. We than have $P^{-1}T = [\tilde{P}_1T_1,\ldots,\tilde{P}_nT_n]$. For maintaining the stability, each T_i and each $\tilde{P}_iT_i$ should be proper. Poles of T_i should be in the open left-half complex plane. Zeros of T_i must cancel the closed right-half plane poles of $\tilde{P}_i$.

(iii) Let $T_i = n_i/d_i$. For tracking step inputs, each $(d_i - n_i)$ should have a zero at $s = 0$, i.e. no constant term.

(iv) $C = -P^{-1}T(1-T)^{-1}$.

Unstable Plants

For unstable plants, similar design procedures can be derived. Again, we are interested in invertible matrices T. This requires the assumption that P is invertible, which in turn implies that N_r and N_ℓ are invertible. As before, we use diagonal T for decoupling and we choose those matrices T such that all terms in I-T have a zero at $s = 0$ for tracking step inputs. For stabilization, however, we choose stable T to make $W = (-N_r^{-1}T + U_r)D_\ell^{-1}$ stable. This process is more involved than the corresponding process for stable plants. The reason for this is that $U_r D_\ell^{-1}$ may be unstable for a given unstable plant. In order to achieve stability, somehow part of $N_r^{-1} T D_\ell^{-1}$ is to be made unstable to cancel the unstable part of $U_r D_\ell^{-1}$. With this in mind, we have the following design procedures for constructing controllers that simultaneously achieve stabilization, decoupling and tracking of step inputs:

(i) For decoupling, choose $T = \text{diag}\{T_1,\ldots,T_n\}$

(ii) Find a stable $T_o = \text{diag}\{T_{01},\ldots,T_{0n}\}$ to make $-N_r^{-1} T_0 D_\ell^{-1} + U_r D_\ell^{-1}$ stable

(iii) Let $T_s = \text{diag}\{T_{s1},\ldots,T_{sn}\}$, $N_r^{-1} = [\tilde{N}_1,\ldots,\tilde{N}_n]$

and $D_\ell^{-1} = \begin{bmatrix} \tilde{D}_i \\ \vdots \\ \tilde{D}_n \end{bmatrix}$, where $\tilde{N}_i$ is the ith column of N_r^{-1} and $\tilde{D}_j$ is the jth row of D_ℓ^{-1}. We then have

$$N_r^{-1} T_s D_\ell^{-1} = \sum_{i=1}^{n} \tilde{N}_i T_{si} \tilde{D}_i .$$

For achieving stability, each T_{si} and $\tilde{N}_i T_{si} \tilde{D}_i$ should be proper. Poles of T_{si} should be in the open left-half complex plane. Zeros of T_{si} must cancel the closed right-half plane poles of $\tilde{N}_i$ and $\tilde{D}_i$.

(iv) Let $T_{oi} = n_{oi}/d_{oi}$ and $T_{si} = n_{si}/d_{si}$.

For tracking step inputs, each $(d_{si}d_{oi} - n_{si}d_{oi} - n_{oi}d_{si})$ should have a zero at $s = 0$, i.e. no constant term.

(v) $T = T_o + T_s$, and

$$C = - PT(I-T)^{-1}.$$

2.3. Complexity of Controllers

As pointed out earlier, the fractional representation approach allows us to search systematically for compensators that achieve various design objectives. This approach, however, does not always bring about simple compensators. As a matter of fact, the time-domain dynamic compensation (ref. 10) is more likely to bring about simple stabilizing compensators than the fractional representation approach. On the other hand, the fractional representation approach is more likely to result in simple stabilizing compensators that also decouple system outputs. The latter is due to the difficulty in dynamic compensation of relating directly the diagonality of a transfer fraction to the formation of the system matrices {A,B,C,D} in the state-space description of a system.

In our design procedures, the compensator C is given by $C = -P^{-1} T(I-T)^{-1}$. For a given plant P, T is to be chosen for forming compensators that stabilize the system, decouple the outputs as well as track step inputs. Under close examination, we notice that the poles of T will basically be cancelled by the same poles of $(I - T)$ in forming C. These poles do not directly affect the complexity of the controllers. However, the total number of the poles determines the degree of freedom in choosing the zeros of T and I-T. For simple compensators, the zeros of T can be chosen to cancel the poles of P^{-1}, and zeros of I-T can be chosen to cancel the zeros of P^{-1}. Overall, T should be kept as simple as possible. The following problem illustrates the details involved. This problem was first discussed in references 6 and 7.

2.4. Design Example

For a plant

$$P(s) = \begin{bmatrix} \frac{1}{s+1} & \frac{s-1}{s+1} \\ 0 & \frac{1}{s-1} \end{bmatrix}$$

We have derived a set of stable matrices N_r, D_r, N_ℓ, D_ℓ, U_r, V_r, U_ℓ and V_ℓ given by

$$D_r = D_\ell = \begin{bmatrix} 1 & 0 \\ 0 & \frac{s-1}{s+2} \end{bmatrix}, \quad N_r = \begin{bmatrix} \frac{1}{s+1} & \frac{(s-1)^2}{(s+1)(s+2)} \\ 0 & \frac{1}{s+2} \end{bmatrix}$$

$$N_\ell = \begin{bmatrix} \frac{1}{s+1} & \frac{s-1}{s+1} \\ 0 & \frac{1}{s+2} \end{bmatrix}, \quad V_r = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$U_r = U_\ell = \begin{bmatrix} 0 & 0 \\ 0 & 3 \end{bmatrix}, \quad V_\ell = \begin{bmatrix} 1 & \frac{-3(s-1)}{s+1} \\ 0 & 1 \end{bmatrix}$$

Since

$$U_r D_\ell^{-1} = \begin{bmatrix} 0 & 0 \\ 0 & \frac{3(s+2)}{s-1} \end{bmatrix}$$

is not stable, a simple stable T_o is to be found to make

$$-N_r^{-1} T_o D_\ell^{-1} + U_r D_\ell^{-1}$$

stable. Such a T_o is given by

$$T_o = \begin{bmatrix} 0 & \text{[illegible]} \\ 0 & \dfrac{9}{s-2)^2} \end{bmatrix}$$

ORIGINAL PAGE IS OF POOR QUALITY

$$-N_r^{-1} T_o D_\ell^{-1} + U_r D_\ell^{-1} = \begin{bmatrix} 0 & 0 \\ 0 & 3 \end{bmatrix}.$$

We then study

$$-N_r^{-1} T_s D_\ell^{-1} = \begin{bmatrix} -T_{s1}(s+1) & T_{s2}(s+2)(s+1) \\ 0 & -T_{s2}\dfrac{(s+2)^2}{s-1} \end{bmatrix}.$$

In order to make this term stable, we must have the following:

1. Let $T_{s1} = n_{s1}/d_{s1}$, so that

   $\deg(n_{s1}) + 1 < \deg(d_{s1})$.

2. Let $T_{s2} = n_{s2}/d_{s2}$, so that

   $\deg(n_{s2}) + 2 < \deg(d_{s2})$.

3. Zeros of d_{s1} and d_{s2} are in the open LHP.

4. Zeros of n_{s1} contain $s = 1$.

Based on points 1 through 4, we have the simplest

$$T_{s1} = a/(s+b)$$

and the simplest

$$T_{s2} = c(s-1)/(s+2)^2(s+d),$$

with positive b and d. For tracking step inputs, both $1 - T_{o1} - T_{s1}$ and $1 - T_{o2} - T_{s2}$ must have a zero at $s = 0$. This reguires that $b - a = 0$ and $4d - 9d + c = 0$. It can be seen that there are many solutions for a, b, c and d. Two sets of solutions are given below, together with the corresponding closed-loop transfer functions and compensators. Choosing $b = 1$ and $d = 1$, we have $a = 1$, $c = 5$ and

$$T = T_o + T_s = \begin{bmatrix} \frac{1}{s+1} & 0 \\ 0 & \frac{14s + 4}{(s+2)^2(s+1)} \end{bmatrix}$$

ORIGINAL PAGE IS OF POOR QUALITY

$$C = \begin{bmatrix} \frac{s+1}{s} & \frac{-(14s+4)(s-1)}{s(s+6)} \\ 0 & \frac{14s + 4}{s(s+6)} \end{bmatrix}.$$

Choosing $b = 1$ and $d = 3$, we have $a = 1$, $c = 15$ and

$$T = \begin{bmatrix} \frac{1}{s+1} & 0 \\ 0 & \frac{24s + 12}{(s+2)^2(s+3)} \end{bmatrix}$$

$$C = \begin{bmatrix} \frac{s+1}{s} & \frac{-3(8s+4)(s-1)}{s(s+8)} \\ 0 & \frac{3(8s+4)}{s(s+8)} \end{bmatrix}.$$

2.5. Hidden Modes

It is known that feedback design using transfer functions may bring about unwanted stable modes hidden in the closed-loop system (ref. 5). In the example of the previous section, the closed-loop system has a transfer function $T(s)$ that corresponds to a fourth order system. However, the plant P is a second order system and the compensator C is a third order system which means that the closed-loop system is actually a fifth order system. The difference in the order of the closed-loop system and its transfer function suggests that there is a hidden mode. The hidden mode in the example is at $s = -1$ which has resulted from the cancellation of the pole of P and the zero of C at $s = -1$. To prevent this type of cancellation, zeros of $I - T$ should be chosen to match the stable poles of P (which are zeros of P^{-1}) in forming C. This selection may prevent us from choosing the simplest T in our design procedures. However, this should not be considered as a setback for finding the simplest compensators, but rather a procedure that guarantees the correct representation of a closed-loop system by its transfer function. With this procedure, the design in the example of the previous section is modified as follows.

The stable pole of P is at $s = -1$. This pole appears in the (1,1) element of P^{-1} as a zero. Hence, $I - T_{o1} = T_{s1} = I - T_{s1}$ should have a zero at $s = -1$ in addition to the zero at $s = 0$ required for tracking step inputs. This requires that $\deg(d_{s1}) \geq 2$. The simplest T_{s1} that has this property is of the form

ORIGINAL PAGE IS
OF POOR QUALITY

$$T_{s1} = \frac{cs + d}{(s+a)(s+b)}$$

with a and $b > 0$. We must have

$$1 - T_{(s)} = \frac{s^2 + (a+b)s + ab - cs - d}{(s+a)(s+b)} = \frac{s(s+1)}{(s+a)(s+b)} .$$

That means $a + b - c = 1$ and $ab - d = 0$. Again, there are many solutions for a, b, c and d. Choosing $a = 3$ and $b = 4$, we have $c = 6$, $d = 12$ and

$$T_{s1} = \frac{6s+12}{(s+3)(s+4)} .$$

Using a set of T_{o2} and T_{s2} as before, we have

$$T = \begin{bmatrix} \frac{6(s+2)}{(s+3)(s+4)} & 0 \\ 0 & \frac{24s+12}{(s+2)^2(s+3)} \end{bmatrix}$$

and

$$C_{(s)} = \begin{bmatrix} \frac{6(s+2)}{s} & \frac{-(24s+12)(s-1)}{s(s+8)} \\ 0 & \frac{24s + 12}{s(s+8)} \end{bmatrix}$$

Note that C remains a third order system and the order of $T_{(s)}$ is 5 which means that there is no longer a hidden mode.

3. EIGENVALUE/EIGENVECTOR ASSIGNMENT PROCEDURES

3.1. Introduction

The design of multivariable feedback control systems using eigenvalue/-eigenvector assignment procedures has received considerable attention during the past several years. Several early studies (refs. 11, 12) focused on an algebraic formulation of the spectral assignment problem. More recent studies (refs. 13-15) have been successful in developing a geometric formulation of this problem. In (ref. 13) the total design freedom available to assign eigenvectors is characterized in terms of eigenspaces. The use of this freedom to achieve desired design specifications has been the subject of an extensive investigation by the current authors and colleagues.

Procedures have been developed for approximating desired mode mixing (ref. 16), reducing eigensystem sensitivity to variations in plant parameters (refs. 17, 18), and reducing the effects of actuator noise on a statistical measure of system performance (ref. 19, 20). In addition, a procedure for modifying the feedback gain matrix to satisfy specified gain constraints (ref. 21, 22) while maintaining a given mode mix has been devised. More recently these procedures have been combined into a single unified design philosophy (ref. 7). This philosophy is reviewed and a computer aided design software package to implement the design philosophy is presented in this section.

3.2. Design Philosophy

The new eigenvalue/eigenvector assignment design philosophy is illustrated in figure 2. The philosophy is based on the premise that achieving a specified set of eigenvalues and approximating a desired set of eigenvectors is of primary importance. Sensitivity reduction, noise suppression, and gain modification are assigned secondary importance and are carried out so as to preserve an initial spectral assignment.

The procedure assumes that the designer is able to identify a desired set of eigenvalues and an approximate set of desired eigenvectors. Eigenvalues directly control the rates of response of the system modes while eigenvectors control how the modes mix among the system states and/or outputs. The design begins with the specification of a desired set of eigenvalues. The procedure realizes arbitrarily specified sets of eigenvalues if the system is controllable. The specified eigenvalues are used to compute the system eigenspaces--the vector spaces in which all realizable system eigenvectors must be contained. These spaces explicitly display the total design freedom available in assigning eigenvectors for a given eigenvalue assignment. Next, the desired set of eigenvectors are specified and projected onto the eigenspaces to locate the set of realizable eigenvectors as close as possible in a minimum square error sense to the desired set of eigenvectors. Since the major advantage of the eigenvalue/eigenvector assignment procedure is the ability to assign eigenvectors, great importance is given to remaining in a small neighborhood of the initial eigenvector assignment.

After the specified eigenvalues have been assigned and the specified eigenvectors have been approximated, the resultant closed-loop system is investigated to determine if all eigenvector components are satisfactory, eigensystem sensitivity is sufficiently low, and gain magnitudes meet specified design constraints. If modification is requried, new eigenvectors are selected in a manner to achieve the desired objective using a gradient search procedure. However, the gradient search is conducted local to the initial eigenvector assignment so that desired mode mixing is retained.

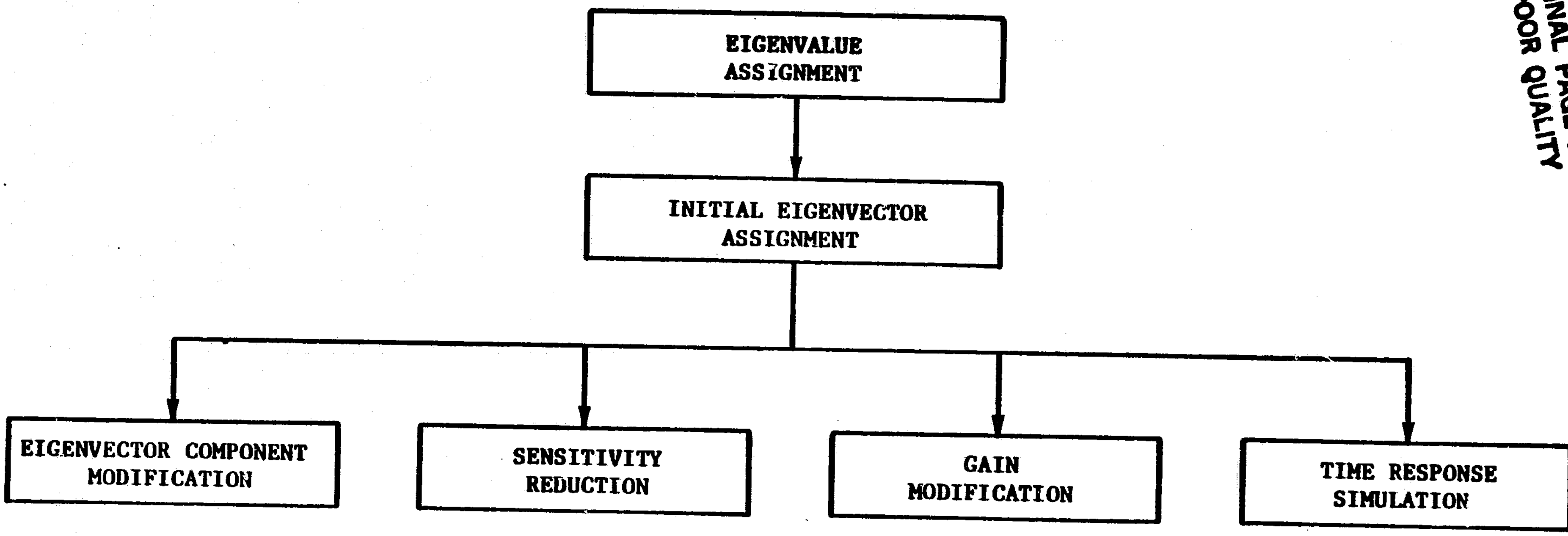


Figure 2. Eigenvalue/eigenvector assignment design philosophy.

3.3. Computer Aided Design Software Package

A flowchart diagram illustrating the organization of the computer software package to implement the eigenvalue/eigenvector assignment procedure is shown in figure 3. The package consists of a number of special purpose subprograms accessible from a main control program. The subprograms can be called in any order to implement specific design objectives, as shown in figure 2. The program is self-instructed and requires no familiarity on the part of the user with the mathematics of spectral assignment.

In the following, the various modes of operation of the program are discussed. An example illustrating the use of the program is presented in the next section, and the program listing is included in the Appendix.

<u>Mode 0</u>

Mode 0 provides a list of references detailing program operation.

<u>Mode 1</u>

Mode 1 is the mode in which system data is entered to the program. Required data includes the number of system states, inputs, and outputs, and the system state variable description in matrix form given by the triple (A, B, C). The user can also set the number of significant digits in user-computer communication as well as the program value for "zero."

<u>Mode 2</u>

In Mode 2, the user specifies desired closed loop system eigenvalues. This mode then internally calculates the corresponding eigenspaces for transmission to other subprograms. User selected eigenvalues are always achieved in this desgin procedure.

<u>Mode 3</u>

The user specifies desired eigenvectors in Mode 3. The program responds with the set of actual closed loop system eigenvectors which are

ORIGINAL PAGE IS
OF POOR QUALITY

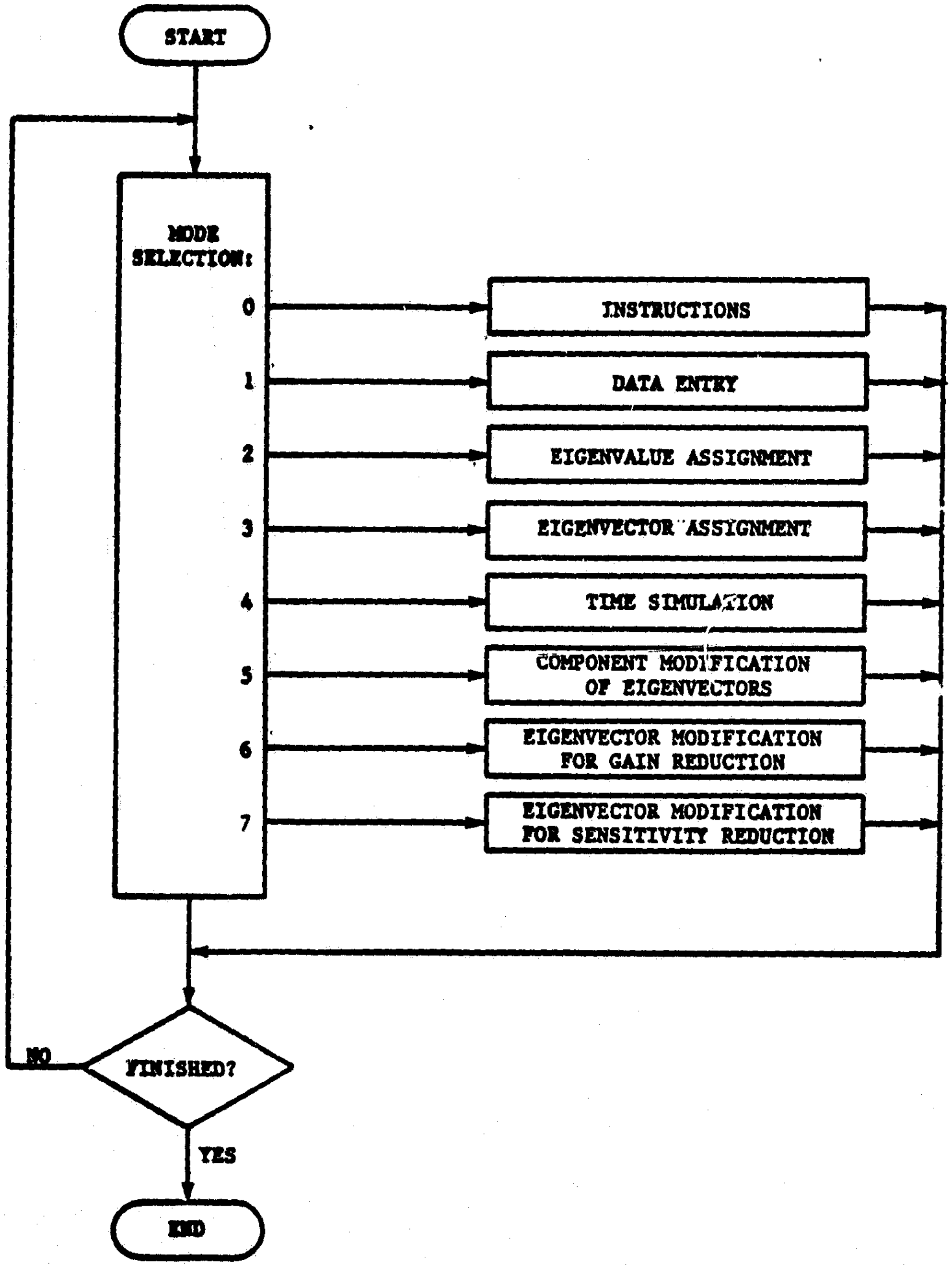


Figure 3. Spectral assignment computer software package organization.

closest to those specified in a least square error sense. The program also displays the error magnitude between each desired and realized eigenvector.

Mode 4

Mode 4 is a closed loop system simulation package. The subprogram numerically solves the set of system state equations subject to user specified inputs and initial conditions. Time responses are plotted separately or on the same axes for comparison.

Mode 5

Mode 5 allows the user to modify specified components in the eigenvector matrix while retaining current values of other components The modification is automatically carried out using a gradient search procedure under the control of the user.

Mode 6

Mode 6 allows the user to modify selected components of the feedback gain matrix while maintaining an approximation to a specified eigenvector matrix. Components of the feedback matrix to be reduced are identified by row and column number. Unequal priority in reducing component magnitudes can be assigned. The modification is automatically conducted by a gradient search algorithm under the control of the user.

Mode 7

In Mode 7 the user can reduce eigensystem sensitivity to variations in plant parameters. The procedure utilizes a gradient search procedure to modify system closed loop eigenvectors to reduce the sensitivity of eignevalues and eigenvectors to changes in specified components of the system state matrices.

3.4. Design Example

In this section an example is presented to illustrate the designer - machine dialog during the design process. Mode 1 is first entered and important system data is input.

Number of states:	3
Number of inputs:	2
Number of outputs:	3
Significant digits:	6
Program zero:	0.0001

ORIGINAL PAGE IS OF POOR QUALITY

$$A = \begin{bmatrix} -2.00 & 0.00 & 1.00 \\ 0.00 & -2.00 & 1.00 \\ 1.00 & 1.00 & -2.00 \end{bmatrix}$$

$$B = \begin{bmatrix} 1.00 & 0.00 \\ 0.00 & 1.00 \\ 0.00 & 0.00 \end{bmatrix}$$

$$C = \begin{bmatrix} 1.00 & 0.00 & 0.00 \\ 0.00 & 1.00 & 0.00 \\ 0.00 & 0.00 & 1.00 \end{bmatrix}$$

Next Mode 2 is entered and desired closed-loop system eigenvalues are input.

$$\lambda_1 = -1.00$$

$$\lambda_2 = -1.20$$

$$\lambda_3 = -3.00$$

Mode 3 is entered next and desired closed-loop system eigenvectors are input. The program responds with the actual set of eigenvectors as close as possible in a least square error sense to those specified. The program also generates the feedback matrix F which assign these eigenvectors and the specified eigenvalues.

$$V_{desired} = \begin{bmatrix} 3.75 & -0.67 & 1.00 \\ 3.25 & 0.75 & -1.00 \\ 7.00 & 0.00 & 0.10 \end{bmatrix}$$

$$V_{actual} = \begin{bmatrix} 3.75 & -0.70 & 0.97 \\ 3.25 & 0.72 & -1.03 \\ 7.00 & 0.02 & 0.07 \end{bmatrix}$$

$$F = \begin{bmatrix} 13.25 & 12.53 & -13.38 \\ -13.16 & -12.45 & 12.30 \end{bmatrix}$$

The designer then enters Mode 4 to simulate the closed-loop system just designed. The user specifies initial conditions and system inputs.

$$x(0) = \begin{bmatrix} 1.00 \\ 0.00 \\ 0.00 \end{bmatrix}$$

$$u(t) = \begin{bmatrix} 0.00 \\ 1.00 \end{bmatrix}$$

The program responds with plots of the system inputs and states shown as functions of time. Plot are also shown for another set of initial conditions and zero input.

$$x(0) = \begin{bmatrix} 1.00 \\ 1.00 \\ 1.00 \end{bmatrix}$$

ORIGINAL PAGE IS
OF POOR QUALITY

$$u(t) = \begin{bmatrix} 0.00 \\ 0.00 \end{bmatrix}$$

It is demonstrated that curves may be requested separately or together for comparison. The designer next enters Mode 5 to modify a component of the eigenvector matrix. The designer specifies that he desires to reduce the magnitude of the (3,1) element of V. Equal weight is assigned to reducing this component and to retaining the current values of other components. After three iterations, a satisfatory V is obtained. The program displays the new feedback gain matrix for this assignment.

$$V = \begin{bmatrix} 3.18 & -0.70 & 0.97 \\ 2.68 & 0.72 & -1.03 \\ 5.85 & 0.02 & 0.07 \end{bmatrix}$$

$$F = \begin{bmatrix} 13.25 & 12.54 & -13.38 \\ -13.16 & -12.45 & 12.29 \end{bmatrix}$$

The designer then returns to Mode 4 to again display the system states.

$$x(0) = \begin{bmatrix} 1.00 \\ 1.00 \\ 1.00 \end{bmatrix}$$

$$u(t) = \begin{bmatrix} 0.00 \\ 0.00 \end{bmatrix}$$

Finally, the designer enters Mode 6 to attempt to reduce the magnitudes of entries in the gain matrix without greatly changing the eigenvector assignment. Equal weight is placed on reducing each component of F. After three iterations a new V matrix and corresponding F matrix are obtained.

$$V = \begin{bmatrix} 3.18 & -0.66 & 0.75 \\ 2.68 & 0.76 & -1.24 \\ 5.85 & 0.16 & 0.48 \end{bmatrix}$$

$$F = \begin{bmatrix} 2.21 & 1.44 & -2.31 \\ -2.12 & -1.41 & 1.25 \end{bmatrix}$$

Not demonstrated here but included in the program is a sensitivity reduction mode. The program is also fully capable of dealing with complex eigenvalue and eigenvector assignments.

```
 ******************************************************************************
********************* SPECTRAL ASSIGNMENT PACKAGE ********************

ENTER DESIRED MODE OF OPERATION,MODE=0,1,2,...,8:
1
************************** MODE 1:DATA ENTRY *************************

**********ENTER OR CHANGE SYSTEM PARAMETERS:

PREVIOUS VALUES?
1
    NS= 3           NI= 2           NO= 3     IDGT= 6      ZERO= 0.000010000000

WISH TO CHANGE?
O
MATRIX A :
            1               2               3

  1  -0.200000E+01   0.000000E+00   0.100000E+01

  2   0.000000E+00  -0.200000E+01   0.100000E+01

  3   0.100000E+01   0.100000E+01  -0.200000E+01
WISH TO CHANGE?
O
MATRIX B :
            1               2

  1   0.100000E+01   0.000000E+00

  2   0.000000E+00   0.000000E+00

  3   0.000000E+00   0.000000E+00
WISH TO CHANGE?
1
ENTER NEW VALUE(S) :
1.000   0.000
0.000   1.000
0.000   0.000
MATRIX C :
            1               2               3

  1   0.100000E+01   0.000000E+00   0.000000E+00

  2   0.100000E+01   0.000000E+00   0.000000E+00

  3   0.000000E+00   0.100000E+01   0.100000E+01
WISH TO CHANGE?
1
ENTER NEW VALUE(S) :
1.000   0.000   0.000
0.000   1.000   0.000
0.000   0.000   1.000
WISH TO EXIT FROM THIS MODE?
1
***************************  EXITING MODE 1  *************************
TERMINATE THIS RUN OR SELECT NEXT MODE:

WISH TO TERMINATE?
O
 *****************************************************************************
********************* SPECTRAL ASSIGNMENT PACKAGE ********************
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
ENTER DESIRED MODE OF OPERATION,MODE=0,1,2,...,8:
2
******************** MODE 2:EIGENVALUE ASSIGNMENT *******************

********** ENTER OR CHANGE EIGENVALUES:


PREVIOUS VALUES?
0
LAMBDA 1:
REAL=    0.000000E+00    IMAG=    0.000000E+00

WISH TO CHANGE?
1
enter new value(s) :
-1.000    0.000
NEXT EIGENVALUE:
PREVIOUS VALUES?
0
LAMBDA 2:
REAL=    0.000000E+00    IMAG=    0.000000E+00

WISH TO CHANGE?
1
enter new value(s) :
-1.2000 0.0000
NEXT EIGENVALUE:
PREVIOUS VALUES?
1
LAMBDA 3:
REAL=   -0.300000E+01    IMAG=    0.000000E+00

WISH TO CHANGE?
0
WISH TO EXIT FROM THIS MODE?
1
***************************  EXITING MODE 2  *************************
TERMINATE THIS RUN OR SELECT NEXT MODE:

WISH TO TERMINATE?
0
 ***********************************************************************
********************* SPECTRAL ASSIGNMENT PACKAGE ********************

ENTER DESIRED MODE OF OPERATION,MODE=0,1,2,...,8:
3
******************** MODE 3:EIGENVECTOR ASSIGNMENT *******************

********** ENTER OR CHANGE EIGENVECTORS:


PREVIOUS VALUES?
0
EIGENVECTOR V 1:    (REAL)              (IMAG)
                 0.000000E+00        0.000000E+00
                 0.000000E+00        0.000000E+00
                 0.000000E+00        0.000000E+00
WISH TO CHANGE?
1
```

ORIGINAL PAGE IS
OF POOR QUALITY

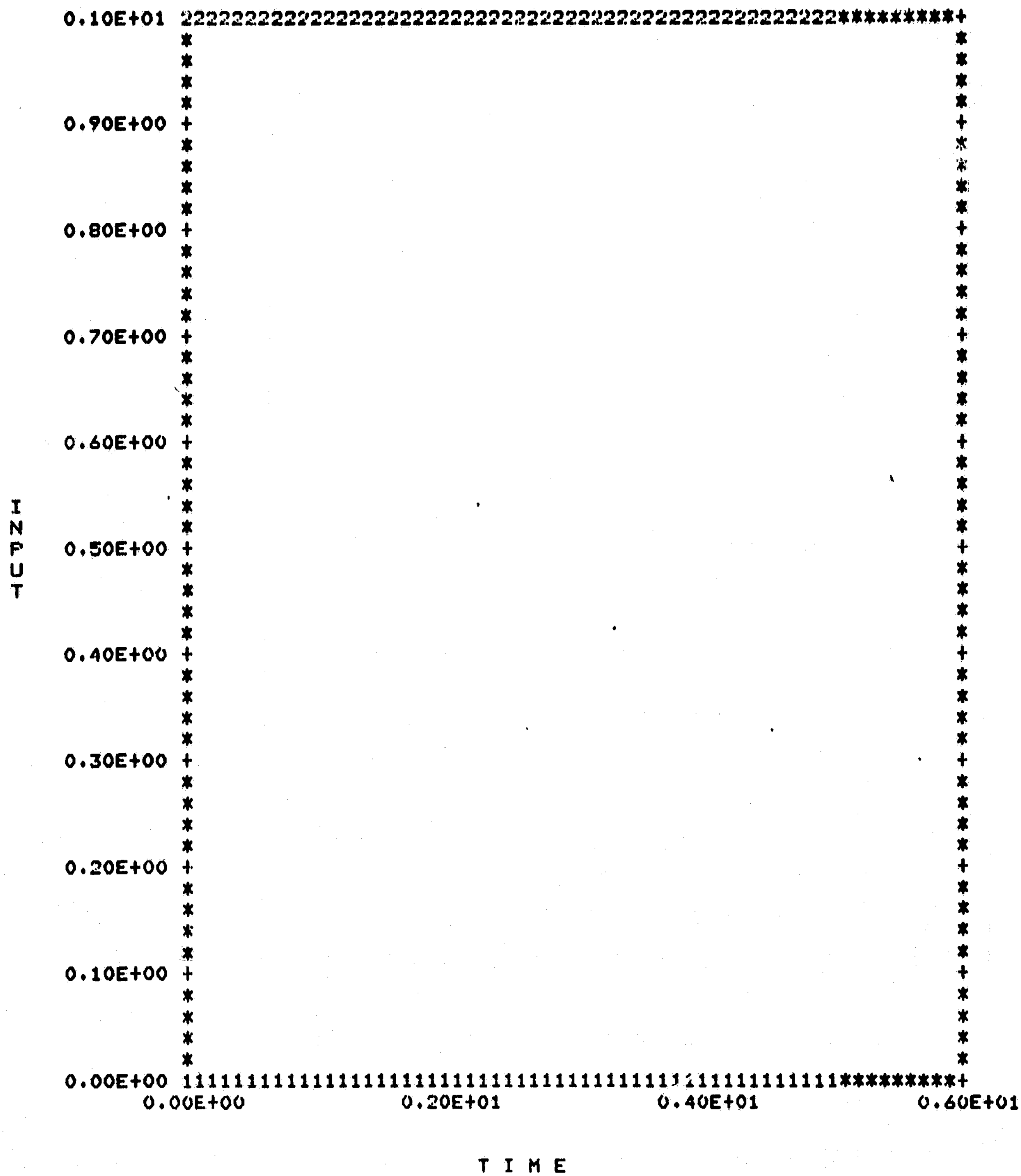
```
ENTER A NEW DESIRED VECTOR :
3.75    0.00
3.25    0.00
7.00    0.00
DESIRED VECTOR:

      0.375000E+01    0.325000E+01    0.700000E+01
ACTUAL VECTORT:

      0.375000E+01    0.325000E+01    0.700000E+01
ERROR VECTORT:

      0.298023E-07    0.298023E-07    0.000000E+00
LENGTH OF THE DESIRED VECTOR   =        8.580501
LENGTH OF THE PROJECTED VECTOR=         8.580501
LENGTH OF THE ERROR VECTOR     =        0.000000
IS THE ERROR ACCEPTABLE?
1
NEXT EIGENVECTOR:
EIGENVECTOR V 2:    (REAL)                 (IMAG)
                 0.000000E+00         0.000000E+00
                 0.000000E+00         0.000000E+00
                 0.000000E+00         0.000000E+00
WISH TO CHANGE?
1
ENTER A NEW DESIRED VECTOR :
-.6700    0.000
.75000    0.000
0.000     0.00
DESIRED VECTOR:

     -0.670000E+00    0.750000E+00    0.000000E+00
ACTUAL VECTORT:

     -0.700303E+00    0.719697E+00    0.242424E-01
ERROR VECTORT:

      0.303030E-01    0.303030E-01   -0.242424E-01
LENGTH OF THE DESIRED VECTOR   =        1.005684
LENGTH OF THE PROJECTED VECTOR=         1.004478
LENGTH OF THE ERROR VECTOR     =        0.049237
IS THE ERROR ACCEPTABLE?
1
NEXT EIGENVECTOR:
EIGENVECTOR V 3:    (REAL)                 (IMAG)
                 0.000000E+00         0.000000E+00
                 0.000000E+00         0.000000E+00
                 0.000000E+00         0.000000E+00
WISH TO CHANGE?
1
ENTER A NEW DESIRED VECTOR :
1.000  0.000
-1.000 0.000
.1000 0.000
DESIRED VECTOR:

      0.100000E+01   -0.100000E+01    0.100000E+00
ACTUAL VECTORT:
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      0.966667E+00  -0.103333E+01   0.666667E-01
ERROR VECTORT:

      0.333333E-01   0.333333E-01   0.333333E-01
LENGTH OF THE DESIRED VECTOR  =      1.417745
LENGTH OF THE PROJECTED VECTOR=      1.416569
LENGTH OF THE ERROR VECTOR    =      0.057735
IS THE ERROR ACCEPTABLE?
1
==========CONTENTS OF 'CURRNT' DATA FILE INCLUDE:
MATRIX V :
              1              2              3

  1   0.375000E+01  -0.700303E+00   0.966667E+00

  2   0.325000E+01   0.719697E+00  -0.103333E+01

  3   0.700000E+01   0.242424E-01   0.666667E-01
WISH TO DISPLAY THE NORMALIZED EIGENVECTORS?
1
NORMALIZED VECTORS :
              1              2              3

  1   0.437037E+00  -0.697181E+00   0.682400E+00

  2   0.378766E+00   0.716489E+00  -0.729462E+00

  3   0.815803E+00   0.241344E-01   0.470621E-01

GAIN MATRIX F:
              1              2              3

  1   0.132526E+02   0.125341E+02  -0.133833E+02

  2  -0.131593E+02  -0.124526E+02   0.122955E+02
MATRIX AHAT:
              1              2              3

  1   0.112526E+02   0.125341E+02  -0.123833E+02

  2  -0.131593E+02  -0.144526E+02   0.132955E+02

  3   0.100000E+01   0.100000E+01  -0.200000E+01
WISH TO EXIT FROM THIS MODE?
1
***************************  EXITING MODE 3  *************************
TERMINATE THIS RUN OR SELECT NEXT MODE:

WISH TO TERMINATE?
0
 ************************************************************************
********************* SPECTRAL ASSIGNMENT PACKAGE ********************

ENTER DESIRED MODE OF OPERATION,MODE=0,1,2,...,8:
4
```

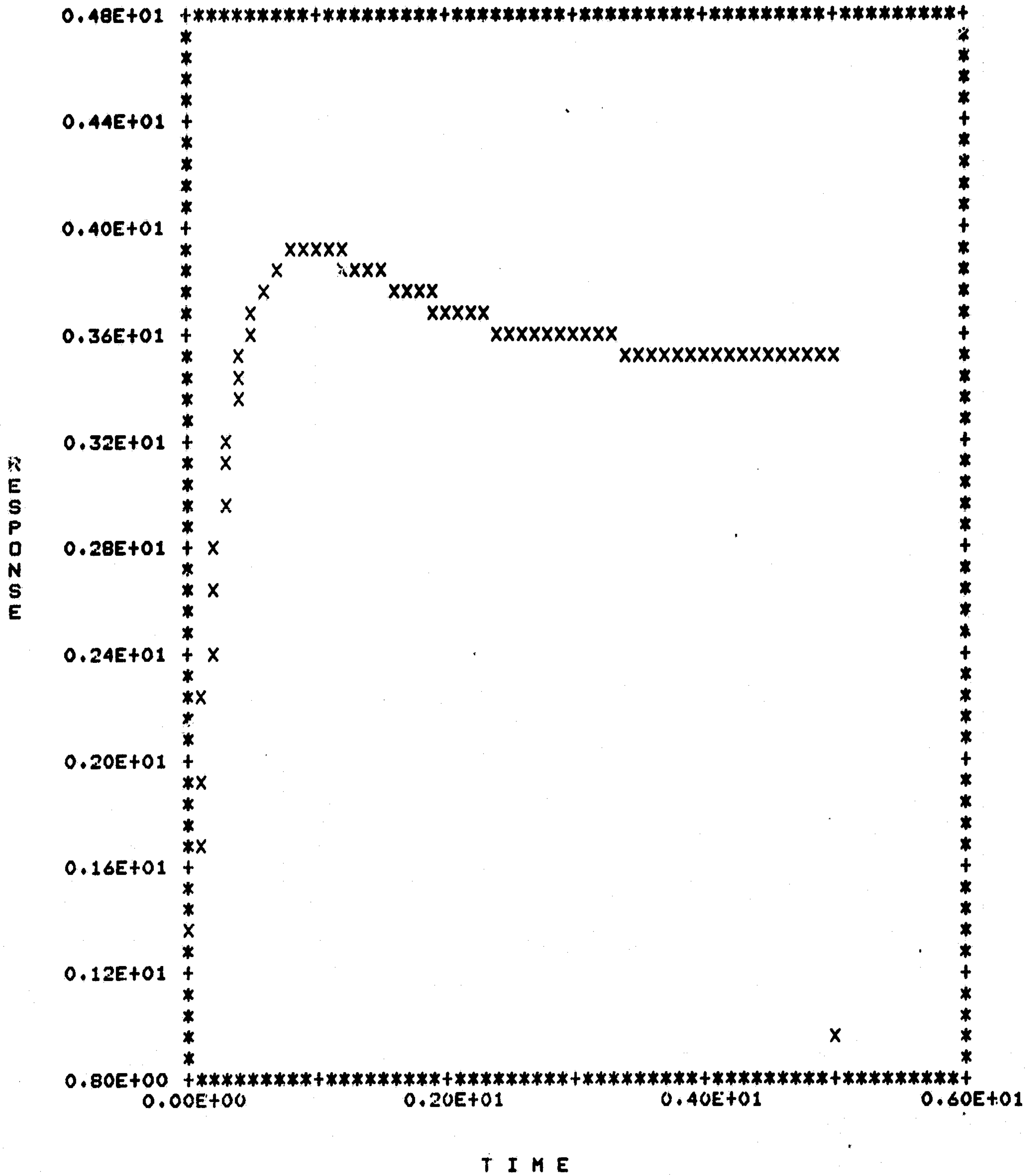
```
*********************** MODE 4:TIME SIMULATION ***********************

********** CHOOSE SIMULATION OPTIONS:
-ENTER: 1 TO SIMULATE [A], 2 TO SIMULATE [AHAT],(3 FOR [ATIL]):
2
ENTER 0 TO SIMULATE OUTPUTS,1 TO SIMULATE STATE VARIABLES:
1
ENTER SIMULATION TIME,(REAL NUMBER IN SECONDS):
5
ENTER NUMBER OF POINTS TO BE CALCULATED,(200 MAX):
150
SPECIFY THE INITIAL CONDITIONS:
X 1(0):
1
X 2(0):
0
X 3(0):
0
CHOOSE INPUT OPTIONS:1 FOR NO INPUT, 2 FOR A STEP INPUT,
                     3 FOR A RAMP,AND 4 FOR A TRUNCATED RAMP:
INPUT OPTION FOR U 1:
1
INPUT OPTION FOR U 2:
2
SPECIFY AMPLITUDE OF THE STEP INPUT U 2:
1
ENTER 0 FOR 80 DISPLAY COLUMNS,1 FOR 129 COLUMNS:
0
ENTER 0 FOR INDIVIDUAL AND 1 FOR MULTIPLE PLOTS:
0
DO YOU WISH TO SET THE MIN-MAX RANGES FOR THE AXES?
0
POSITION PAPER AT TOP OF FORM AND TYPE ANY INTEGER
YOU MAY ADD A SHORT NOTE (20 CHARACTERS.)
1 HERE WE GO
```

SYSTEM INPUTS



POSITION PAPER AT TOP OF FORM AND TYPE ANY INTEGER
YOU MAY ADD A SHORT NOTE (20 CHARACTERS.)
1

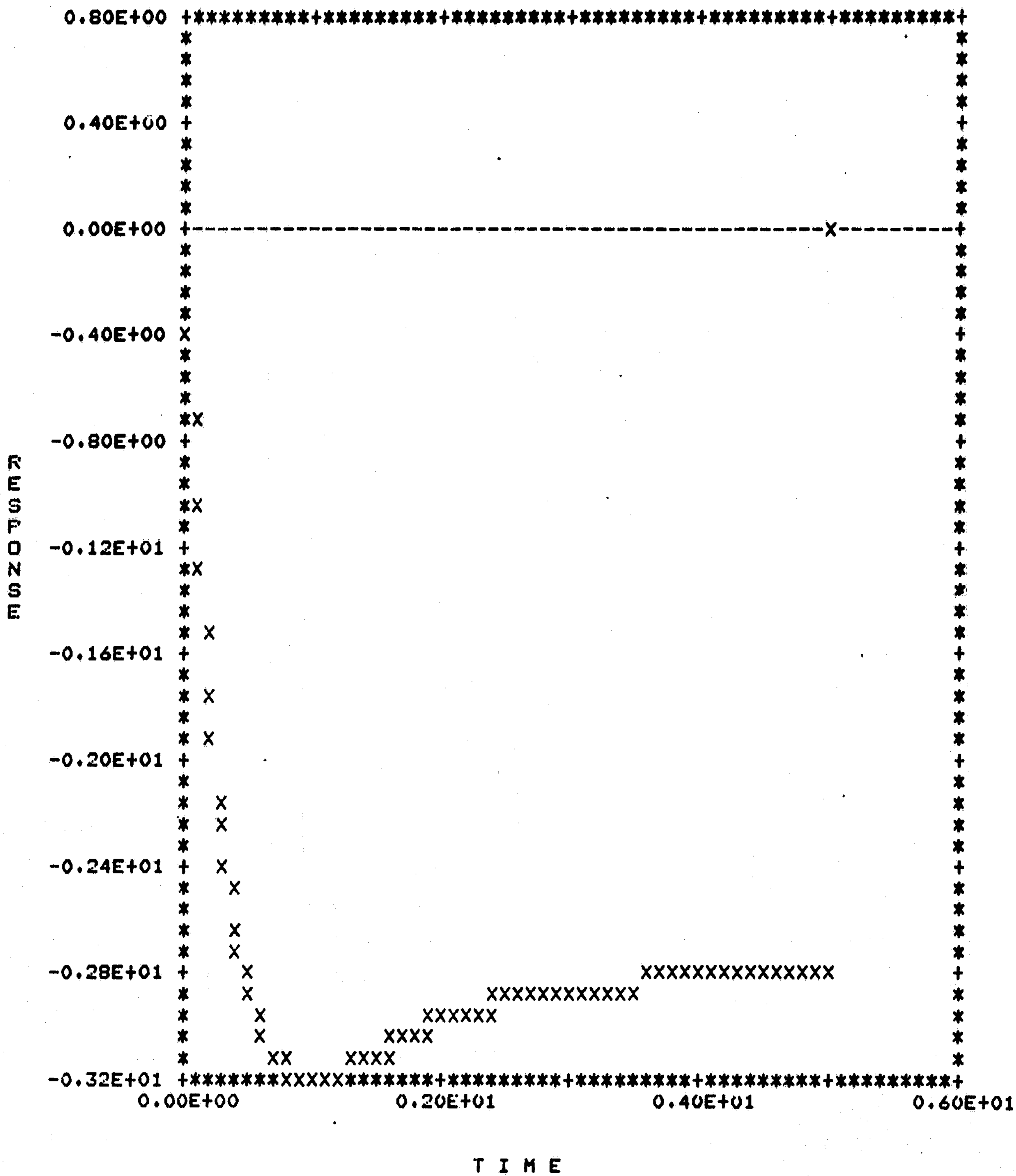
TIME SIMULATION



POSITION PAPER AT TOP OF FORM AND TYPE ANY INTEGER
YOU MAY ADD A SHORT NOTE (20 CHARACTERS.)
1

ORIGINAL PAGE IS
OF POOR QUALITY

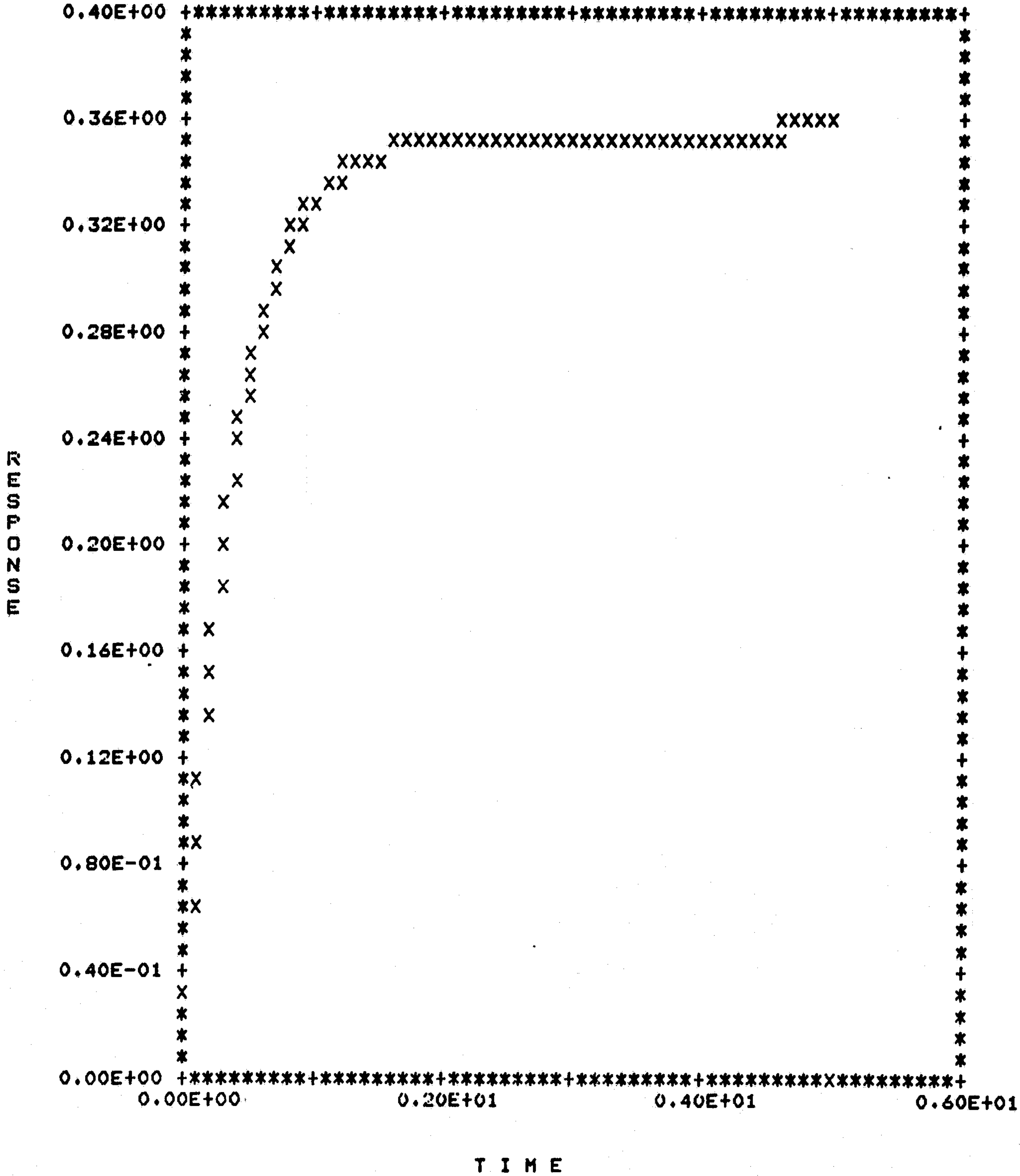
TIME SIMULATION



POSITION PAPER AT TOP OF FORM AND TYPE ANY INTEGER
YOU MAY ADD A SHORT NOTE (20 CHARACTERS.)
1

ORIGINAL PAGE IS
OF POOR QUALITY

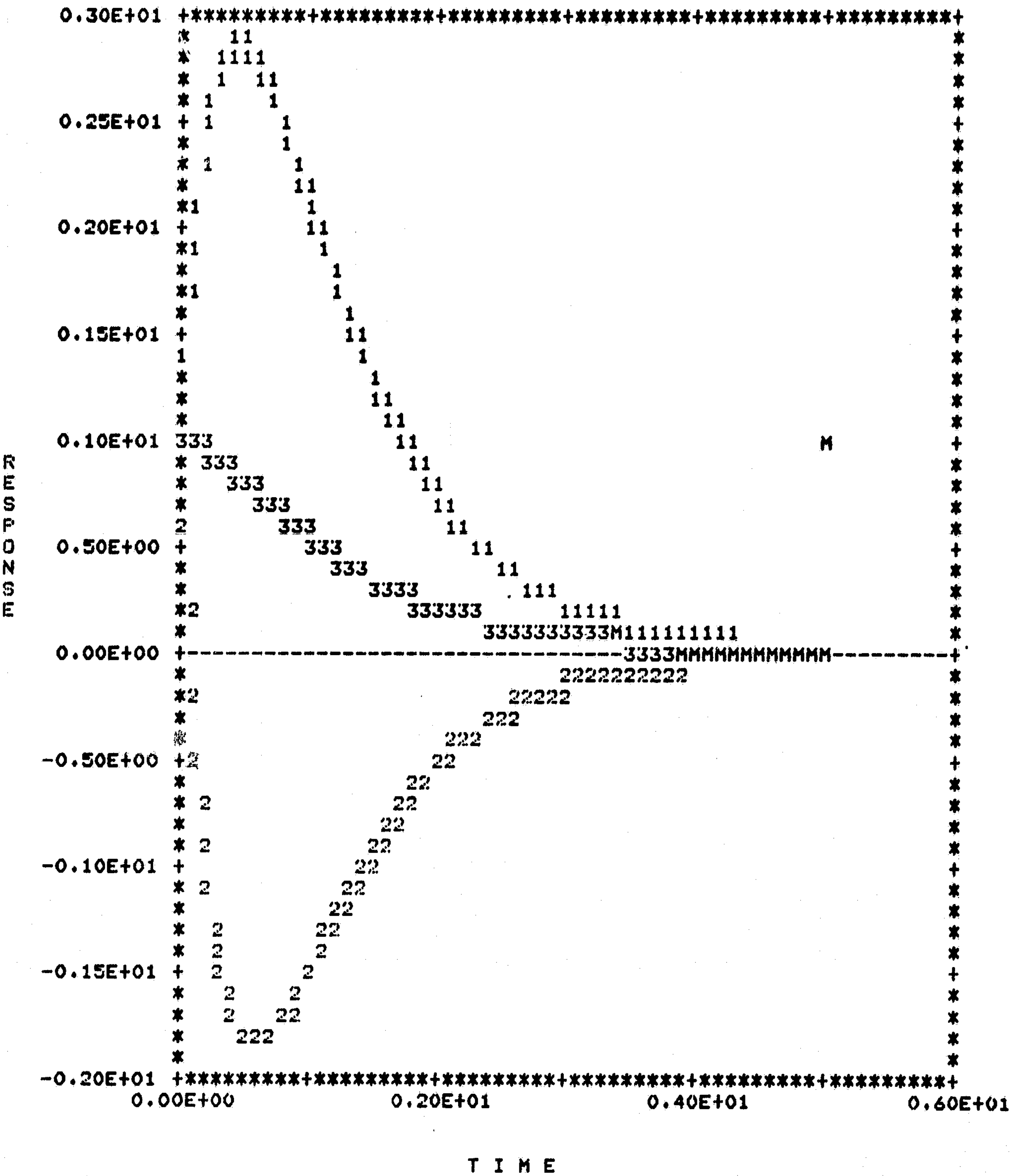
TIME SIMULATION



WISH TO REPEAT THE PLOTTING?
0
WISH TO EXIT FROM THIS MODE?
0

```
*********************** MODE 4:TIME SIMULATION *******************

********** CHOOSE SIMULATION OPTIONS:
-ENTER: 1 TO SIMULATE [A], 2 TO SIMULATE [AHAT],(3 FOR [ATIL]):
2
ENTER 0 TO SIMULATE OUTPUTS,1 TO SIMULATE STATE VARIABLES:
1
ENTER SIMULATION TIME,(REAL NUMBER IN SECONDS):
5
ENTER NUMBER OF POINTS TO BE CALCULATED,(200 MAX):
150
SPECIFY THE INITIAL CONDITIONS:
X 1(0):
1
X 2(0):
1
X 3(0):
1
CHOOSE INPUT OPTIONS:1 FOR NO INPUT, 2 FOR A STEP INPUT,
                     3 FOR A RAMP,AND 4 FOR A TRUNCATED RAMP:
INPUT OPTION FOR U 1:
1
INPUT OPTION FOR U 2:
1
ENTER 0 FOR 80 DISPLAY COLUMNS,1 FOR 129 COLUMNS:
0
ENTER 0 FOR INDIVIDUAL AND 1 FOR MULTIPLE PLOTS:
1
DO YOU WISH TO SET THE MIN-MAX RANGES FOR THE AXES?
0
POSITION PAPER AT TOP OF FORM AND TYPE ANY INTEGER
YOU MAY ADD A SHORT NOTE (20 CHARACTERS.)
1
```

TIME SIMULATION




```
WISH TO REPEAT THE PLOTTING?
[SYSTEM] This Job will be killed in 4 minutes if it remains inactive
0
WISH TO EXIT FROM THIS MODE?
1
**************************  EXITING MODE 4  *************************
```

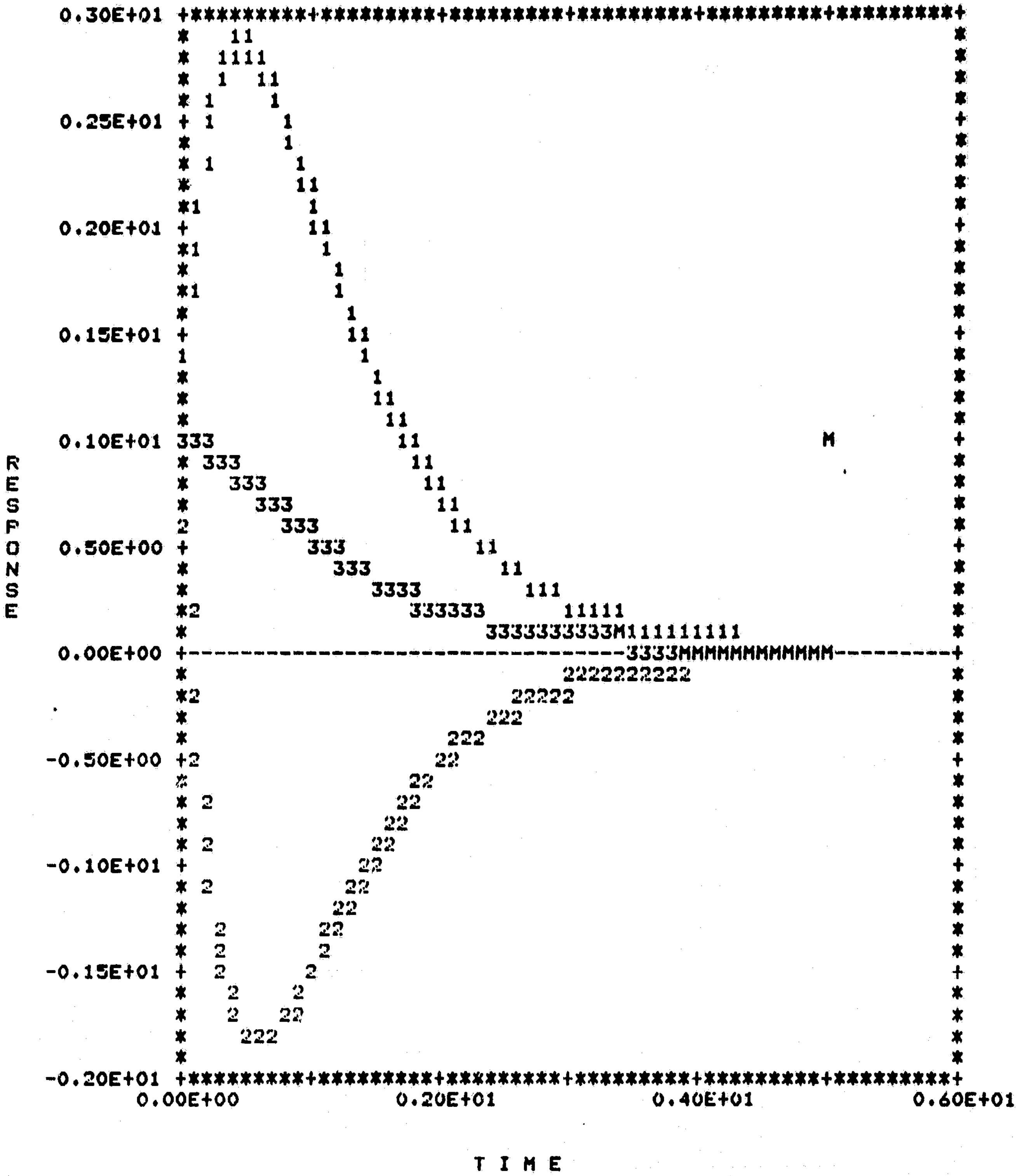
ORIGINAL PAGE IS
OF POOR QUALITY

```
 ***************************************************************************
 ********************* SPECTRAL ASSIGNMENT PACKAGE ********************

 ENTER DESIRED MODE OF OPERATION,MODE=0,1,2,...,8:
 5
********************** MODE 5:COMPONENT REDUCTION *******************

ENTER THE COORDINATES OF THE COMPONENT TO BE REDUCED
ROW=--,COLUMN=--(BOTH INTEGERS):
3  1
SET DESIRED WEIGHTS,DEFAULT VALUES ARE:
F1=F2=1.000
WISH TO CHANGE?
0
                         J1=    0.490000E+02       J2=    0.000000E+00
COST=   0.490000E+02
GRADIENT MATRIX:
           1                  2                  3

  1  -0.707107E+00    0.000000E+00    0.000000E+00

  2  -0.707107E+00    0.000000E+00    0.000000E+00
GRADIENT SEARCH ROUTINE,SET SEARCH PARAMETERS:

Default values are:
# of steps,N=  1    step size,d=    0.100000E-01    dmin=    0.100000E-04

Wish to change?
0
                         J1=    0.488022E+02       J2=    0.999999E-04
NEW COST=   0.488023E+02

Cost Function=    0.488023E+02
Wish to continue the search?
1
GRADIENT SEARCH ROUTINE,SET SEARCH PARAMETERS:

Default values are:
# of steps,N=  1    step size,d=    0.100000E-01    dmin=    0.100000E-04

Wish to change?
1
Enter new values:
1  0.5 .0001
                         J1=    0.394227E+02       J2=    0.260100E+00
NEW COST=   0.396828E+02

Cost Function=    0.396828E+02
Wish to continue the search?
1
GRADIENT SEARCH ROUTINE,SET SEARCH PARAMETERS:

Default values are:
# of steps,N=  1    step size,d=    0.500000E+00    dmin=    0.100000E-03

Wish to change?
1
Enter new values:
1   .3  .0001
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
                    J1=    0.342750E+02        J2=    0.656100E+00
NEW COST=   0.349311E+02
 Cost Function=   0.349311E+02
 Wish to continue the search?
 0
 MATRIX V :
             1              2              3

   1   0.317724E+01  -0.700303E+00   0.966667E+00

   2   0.267724E+01   0.719697E+00  -0.103333E+01

   3   0.585449E+01   0.242424E-01   0.666667E-01
WISH TO DISPLAY THE NORMALIZED EIGENVECTORS?
0
GAIN MATRIX F:
             1              2              3

   1   0.132540E+02   0.125355E+02  -0.133827E+02

   2  -0.131608E+02  -0.124540E+02   0.122949E+02
TERMINATE THIS RUN OR SELECT NEXT MODE:

WISH TO TERMINATE?
1

*********************** MODE 4:TIME SIMULATION ***********************

********** CHOOSE SIMULATION OPTIONS:
-ENTER: 1 TO SIMULATE [A], 2 TO SIMULATE [AHAT],(3 FOR [ATIL]):
2
ENTER 0 TO SIMULATE OUTPUTS,1 TO SIMULATE STATE VARIABLES:
1
ENTER SIMULATION TIME,(REAL NUMBER IN SECONDS):
5
ENTER NUMBER OF POINTS TO BE CALCULATED,(200 MAX):
150
SPECIFY THE INITIAL CONDITIONS:
X 1(0):
1
X 2(0):
1
X 3(0):
1
CHOOSE INPUT OPTIONS:1 FOR NO INPUT, 2 FOR A STEP INPUT,
                    3 FOR A RAMP,AND 4 FOR A TRUNCATED RAMP:
INPUT OPTION FOR U 1:
1
INPUT OPTION FOR U 2:
1
ENTER 0 FOR 80 DISPLAY COLUMNS,1 FOR 129 COLUMNS:
0
ENTER 0 FOR INDIVIDUAL AND 1 FOR MULTIPLE PLOTS:
1
DO YOU WISH TO SET THE MIN-MAX RANGES FOR THE AXES?
0
POSITION PAPER AT TOP OF FORM AND TYPE ANY INTEGER
YOU MAY ADD A SHORT NOTE (20 CHARACTERS.)
1
```

TIME SIMULATION



WISH TO REPEAT THE PLOTTING?
0\0\0
WISH TO EXIT FROM THIS MODE?
1
************************** EXITING MODE 4 *************************

ORIGINAL PAGE IS
OF POOR QUALITY

```
 ***************************************************************************
 ********************** SPECTRAL ASSIGNMENT PACKAGE ********************

 ENTER DESIRED MODE OF OPERATION,MODE=0,1,2,...,8:
 6
 MATRIX V :
                 1                  2                  3

   1    0.317724E+01   -0.700303E+00    0.966667E+00

   2    0.267724E+01    0.719697E+00   -0.103333E+01

   3    0.585449E+01    0.242424E-01    0.666667E-01
 GAIN MATRIX F:
                 1                  2                  3

   1    0.132540E+02    0.125355E+02   -0.133827E+02

   2   -0.131608E+02   -0.124540E+02    0.122949E+02
 ********************** MODE 6:GAIN REDUCTION *************************

 SET ALPHA PARAMETERS :
 DEFAULT VALUES ARE :
 GAIN PARAMETERS :
                 1                  2                  3

   1    0.100000E+01    0.100000E+01    0.100000E+01

   2    0.100000E+01    0.100000E+01    0.100000E+01
 WISH TO CHANGE:
 0
 COST=    0.991380E+03
 Gradient matrix:
                 1                  2                  3

   1   -0.328984E-04    0.123434E+00   -0.719359E+00

   2    0.390425E-04    0.120108E+00   -0.672949E+00
 GRADIENT SEARCH ROUTINE,SET SEARCH PARAMETERS:

 Default values are:
 # of steps,N=  1    step size,d=    0.100000E-01    dmin=    0.100000E-04

 Wish to change?
 1
 Enter new values:
 3
 .1  .0001
 NEW COST=    0.107344E+03
 NEW COST=    0.389868E+02
 NEW COST=    0.203202E+02
 Cost Function=    0.203202E+02
 Wish to continue the search?
 0
 MATRIX V :
                 1                  2                  3

   1    0.317723E+01   -0.663273E+00    0.750859E+00

   2    0.267726E+01    0.755729E+00   -0.123522E+01
```

```
  3   0.585449E+01    0.115571E+00    0.484359E+00
WISH TO DISPLAY THE NORMALIZED EIGENVECTORS?
0
GAIN MATRIX F:
              1                  2                  3

  1   0.220673E+01    0.143515E+01   -0.231119E+01

  2  -0.212192E+01   -0.140673E+01    0.125216E+01
TERMINATE THIS RUN OR SELECT NEXT MODE:

WISH TO TERMINATE?
1
```

REFERENCES

1. D. C. Youla, J. J. Bongiorno, Jr., and H.A. Jabr, "Modern Wiener-Hopf Design of Optimal Controllers - Part I: The Single-Input-Output Case." IEEE Trans. on Automatic Control, Vol. AC-21, February 1976.

2. D. C. Youla, H. A. Jabr, and J. J. Bongiorno, Jr., "Modern Wiener-Hopt Design of Optimal Controllers - Part II: The Multivariable Case." IEEE Trans. on Automatic Control, Vol. AC-21, June 1976.

3. C. A. Desoer, R. W. Lieu, J. Murray, and R. Saeks, " Feedback System Design: The Fractional Representation Approach to Analysis and Synthesis." IEEE Trans. on Automatic Control, Vol. AC-25, June 1980.

4. J. L. Peczkowski and M. K. Sain, "Linear Multivariable Synthesis with Transfer Functions," in Alternatives for Linear Multivariable Control, edited by Sain, Peczkowski and Melsa, NEC, Chicago, 1978.

5. P. J. Antsaklis and M. K. Sain, "Unity Feedback Compensation of Unstable Plants." Proceedings of IEEE Conference on Decision and Control, December 1981.

6. R. R. Mielke and L. J. Tung, "Design of Multivariable Feedback Control System Via Spectral Assignment." Progress Report, NASA Grant NSG-1650, Old Dominion University Research Foundation, October 1981.

7. R. R. Mielke, L. J. Tung and M. Marefat, "Design of Multivariable Feedback Control System Via Spectral Assignment." Progress Report, NASA Grant NSG-1650, Old Dominion University Research Foundation, June 1982.

8. R. Saeks and J. Murray, "Feedback System Design: The Tracking and Disturbance Rejection Problems." IEEE Transactions on Automatic Control, Vol. AC-26, February 1981.

9. R. R. Mielke, L. J. Tung, and S. R. Liberty, "Design of Multivariable Feedback Control Systems Via Spectral Assignment." Progress Report NASA Grant NSG-1650, Old Dominion University Research Foundation, April 1981.

10. F. M. Brasch, Jr., and J. B. Pearson, "Pole Placement Using Dynamic Compensators.", IEEE Trans. on Automatic Control, Vol. AC-15, No. 1, February 1970.

11. S. Srinathkumar: "Spectral Characterization of Multi-Input Dynamic Systems." Ph.D. dissertation, Oklahoma State University, Stillwater, OK, May 1976.

12. S. Srinathkumar: "Eigenvalue/Eigenvector Assignment Using Output Feedback." IEEE Transactions on Automatic Control, Vol. AC-23, February 1978, pp. 79-81.

13. B.C. Moore: "On the Flexibility Offered by State Feedback in Multivariable Systems Beyond Closed-Loop Eigenvalue Assignment." IEEE Transactions on Automatic Control, Vol. AC-21, October 1976, pp. 685-692.

14. H. Kumura: "A Further Result on the Problem of Pole Assignment by Output Feedback." IEEE Transactions on Automatic Control, Vol. AC-22, June 1977, pp. 458-463.

15. E. J. Davison and S. H. Wang: "On POle Assignment in Linear Multivariable Systems." IEEE Transactions on Automatic Control, Vol. AC-20, August 1975, pp. 516-518.

16. R. A. Maynard, R. R. Mielke, S. R. Liberty, and S. Srinathkumar: "A New Spectral Synthesis Procedure for Multivariable Regulators." Allerton Conference on Communication, Control, and Computing, Vol. 16, October 1978.

17. R. R. Mielke, R. A. Maynard, S. R. Liberty, and S. Srinathkumar: "Design of Robust State Feedback Controllers Via Eigenvalue/Eigenvector Assignment." Asilomar Conference on Circuits, Systems, and Computers, Vol. 12, November 1978.

18. S. R. Liberty, R. R. Mielke, and R. A. Maynard: "A Summary of Spectral Synthesis Procedures for Multivariable Systems." Final Report, NASA Grant NSG-1519, Old Dominion University Research Foundation, November 1979.

19. R. A. Maynard, R. R. Mielke, and S. R. Liberty: "Eigenvector Assignment for Noise Suppression." Southeastern Symposium on System Theory, Vol. 13, March 1981.

20. S. R. Liberty, R. R. Mielke, and R. A. Maynard: "Design of Multivariable Feedback Control Systems Via Spectral Assignment." Progress Report, NASA Grant NSG-1650, Old Dominion University Research Foundation, October 1980.

21. R. R. Mielke and L. J. Tung: "Design of Multivariable Feedback Control Systems Via Spectral Assignment." Progress Report, NASA Grant NSG-1650, Old Dominion University Research Foundation, October 1981.

22. T. M. Owens and R. R. Mielke: "A Spectral Assignment Gain Modification Procedure," Proceedings of Southeastern Symposium on System Theory, March 1981.

APPENDIX

Eigenvalue/Eigenvector Assignment Program Listing

```
C*************************************************************
C*************************************************************
C-Function: Mode Selection.
C-IMSL routines called: UGETIO.
C-Spectral Assignment routines: MODE0 through MODE8.
C-Logical devices; Input Unit:   5   Output Unit:   5
C             Storage Unit(s): IU=20
C-Random Access Files: SYSTEM.DAT
        REAL A(10,10),B(10,10),C(10,10),ZERO
        INTEGER MODE,IDGT,NS,NI,NO
        COMMON/SYS/A,B,C,ZERO,IDGT,NS,NI,NO
        CALL UGETIO (3,5,5)
C*****************************************************************
        IRS=102
        IU=20
      OPEN (FILE='SYSTEM.DAT',ACCESS='RANDOM',RECORD SIZE=IRS
     1,UNIT=IU,MODE='BINARY',DEVICE='DSK',DISPOSE='SAVE')
  100 WRITE (5,101)
  101   FORMAT (1H/,1X,70(1H*),/,1X,21(1H*),
     129H SPECTRAL ASSIGNMENT PACKAGE ,20(1H*),//
     2,1X,49HENTER DESIRED MODE OF OPERATION,MODE=0,1,2,...,8:)
        READ (5,*) MODE
        IF (MODE.LE.0) GO TO 80
        GO TO (1,2,3,4,5,6,7,8),MODE
    1 CALL MODE1
        GO TO 99
    2 CALL MODE2
        GO TO 99
    3  CALL MODE3
        GO TO 99
    4 CALL MODE4
        GO TO 99
    5   CALL MODE5
        GO TO 99
    6   CALL MODE6
        GO TO 99
    7   CALL MODE7
        GO TO 99
    8   CALL MODE8
        GO TO 99
   80 CALL MODE0
   99 WRITE (5,102)
  102 FORMAT (1X,39HTERMINATE THIS RUN OR SELECT NEXT MODE:,//
     1,1X,18HWISH TO TERMINATE?)
        READ (5,*) I
        IF (I.LE.0) GO TO 100
        STOP
        END
```

```
C*********************************************************************
C*********************************************************************
      SUBROUTINE MODE1
C-Function: System data entry.
C-IMSL routines called: USWFM.
C-Spectral Assignment routines: -
C5Logical devices; Input Unit:    5   Output Unit:    5
C                Storage Unit(s): IU=20.
C-Random Access Files: SYSTEM.DAT .
      REAL A(10,10),B(10,10),C(10,10),NULL(5)
      COMMON/SYS/A,B,C,ZERO,IDGT,NS,NI,NO
      IU=20
  210 WRITE (5,1)
    1 FORMAT (1X,26(1H*),19H MODE 1:DATA ENTRY ,25(1H*),//,1X,10(1H*)
     1,34HENTER OR CHANGE SYSTEM PARAMETERS:,//)
      WRITE (5,4)
    4 FORMAT (1X,16HPREVIOUS VALUES?)
      READ (5,*) I7
      IF (I7.GT.0) GO TO 220
  230 WRITE (5,2) NS,NI,NO,IDGT,ZERO
    2 FORMAT (5X,3HNS=,I2,10X,3HNI=,I2,10X,3HNO=,I2
     1,5X,5HIDGT=,I2,5X,5HZERO=,F15.12,//,1X,15HWISH TO CHANGE?)
      READ (5,*) I1
      IF (I1.LE.0) GO TO 100
      WRITE (5,5)
    5 FORMAT (1X,20HENTER NEW VALUE(S) :)
      READ (5,*) NS,NI,NO,IDGT,ZERO
      WRITE (IU'1) NS,NI,NO,IDGT,ZERO
C
  100 CALL USWFM (10HMATRIX A :,10,A,10,NS,NS,4)
      WRITE (5,3)
    3 FORMAT (1X,15HWISH TO CHANGE?)
      READ (5,*) I2
      IF (I2.LE.0) GO TO 130
      WRITE (5,5)
      READ (5,*) ((A(I,J),J=1,NS),I=1,NS)
      WRITE (IU'2) ((A(I,J),J=1,NS),I=1,NS)
C
  130 CALL USWFM (10HMATRIX B :,10,B,10,NS,NI,4)
      WRITE (5,3)
      READ (5,*) I3
      IF (I3.LE.0) GO TO 160
      WRITE (5,5)
      READ (5,*) ((B(I,J),J=1,NI),I=1,NS)
      WRITE (IU'3) ((B(I,J),J=1,NI),I=1,NS)
C
  160 CALL USWFM (10HMATRIX C :,10,C,10,NO,NS,4)
      WRITE (5,3)
      READ (5,*) I4
      IF (I4.LE.0) GO TO 200
      WRITE (5,5)
      READ (5,*) ((C(I,J),J=1,NS),I=1,NO)
      WRITE (IU'4) ((C(I,J),J=1,NS),I=1,NO)
      GO TO 200
C*******THIS BLOCK  ACCESSED ONLY BY A GO TO 220 STATEMENT***
  220 CONTINUE
```

```
      READ (IU'1) NS,NI,NO,IDGT,ZERO
      READ (IU'2) ((A(I,J),J=1,NS),I=1,NS)
      READ (IU'3) ((B(I,J),J=1,NI),I=1,NS)
      READ (IU'4) ((C(I,J),J=1,NS),I=1,NO)
      GO TO 230
C          ***********************************
  200 WRITE (5,6)
    6 FORMAT (1X,29HWISH TO EXIT FROM THIS MODE? )
      READ (5,*) I6
      IF (I6.LE.0) GO TO 210
      WRITE (5,7)
    7 FORMAT (1X,27(1H*),18H  EXITING MODE 1  ,25(1H*))
      RETURN
      END
```

```
C*****************************************************************
C*****************************************************************
        SUBROUTINE MODE2
C-Function: Eigenvalue Assignment.
C-IMSL routines called: (USWFM).
C-Spectral Assignment routines: NSA,TRANS .
C-Logical devices; Input Unit:   5   Output Unit:   5
C              Storage Unit(s): IU=20,IU=20+I for I=1,NS.
C-Random Access Files: SYSTEM.DAT,FORxx.DAT where xx=IU=20+I for I=1,NS.
        REAL LRE(10),LIM(10),S(10,30),SCOPY(10,30),SP(10,10),SPP(10,20)
        REAL X(30,20),ML(10,10),NL(10,10)
        REAL NLC(10,20),PLC(10,20),MLC(10,20)
        REAL ALPHA(20,20),BETA(20,20),KA(20,10),KB(20,10),GAMA(20,20)
        REAL ACOPY(20,20),AP(20,20),APP(20,10)
        REAL STAR(20,20),QL(10,20),RL(10,20)
        REAL A(10,10),B(10,10),C(10,10)
        COMMON/SYS/A,B,C,ZERO,IDGT,NS,NI,NO
        COMMON/NSPA/ML,NL,NLC,PLC,MLC,STAR,QL,RL/EIG/LRE,LIM
C****************** READ SYSTEM DATA *************************
        IRS=102
        IU=20
        READ (IU'1) NS,NI,NO,IDGT,ZERO
        READ (IU'2) ((A(I,J),J=1,NS),I=1,NS)
        READ (IU'3) ((B(I,J),J=1,NI),I=1,NS)
C
 910    I=1
        WRITE (5,1)
   1  FORMAT (1X,20(1H*),30H MODE 2:EIGENVALUE ASSIGNMENT ,20(1H*),//
     1,1X,10(1H*),29H ENTER OR CHANGE EIGENVALUES:,//)
C       WRITE (5,33) ZERO,IDGT                    !**
C 33    FORMAT (1X,5HZERO=,F15.12,1X,5HIDGT=,I2)  !**
 999  CONTINUE
        IU=I+20
        IRS=202
      OPEN (ACCESS='RANDOM',RECORD SIZE=IRS
     1,UNIT=IU,MODE='BINARY',DEVICE='DSK',DISPOSE='SAVE')
C
        WRITE (5,11)
 11     FORMAT (1X,16HPREVIOUS VALUES?)
        READ (5,*) KO
        IF (KO.GT.O) GO TO 12
        GO TO 13
C
 12     READ (IU'1) LRE(I),LIM(I)
 13     WRITE (5,2) I,LRE(I),LIM(I)
   2  FORMAT (1X,6HLAMBDA,I2,1H:,/,1X,5HREAL=,E15.6,2X,6H IMAG=
     1,E15.6,//,1X,15HWISH TO CHANGE?)
        READ (5,*) K1
        IF (K1.LE.0) GO TO 50
        write (5,14)
  14    format (1x,20henter new value(s) :)
        READ (5,*) LRE(I),LIM(I)
        WRITE (IU'1) LRE(I),LIM(I)
C************************IS LAMBDA-I REAL OR COMPLEX?*************
        IF (ABS(LIM(I)).GT.ABS(ZERO)) GO TO 100
C************************REAL NULL SPACE FORMULATION**************
```

```
C*************************FORM S-LAMBDA-I,(NSX(NS+NI))*************
          DO 10 II=1,NS
          DO 10 IJ=1,NS
          S(II,IJ)=-A(II,IJ)
          IF (II.EQ.IJ) S(II,IJ)=S(II,IJ)+LRE(I)
   10     CONTINUE
        INS=NS+1
        JNS=NS+NI
          DO 20 II=1,NS
          DO 20 IJ=INS,JNS
        IDUM=IJ-NS
          S(II,IJ)=B(II,IDUM)
   20     CONTINUE
C         CALL USWFM (11HMATRIX SLI:,11,S,10,NS,JNS,4)  !**
C*************************CALL NSA*******************************
C         WRITE (5,3)                               !**
C   3     FORMAT (1X,'NULL SPACE OF S-LAMBDA-I,X=KL')       !**
C         write (5,33)  zero,idgt                 !**
          CALL NSA(NS,JNS,S,10,30,X,30,20,ZERO,IDGT,SCOPY,SPP,SP)
C*******************PARTITION X=KL INTO NL AND ML ************
          DO 30 II=1,NS
          DO 30 IJ=1,NI
          NL(II,IJ)=X(II,IJ)
   30     CONTINUE
          WRITE (IU'3) ((NL(II,IJ),IJ=1,NI),II=1,NS)
          DO 40 II=INS,JNS
          DO 40 IJ=1,NI
          IML=II-NS
          ML(IML,IJ)=X(II,IJ)
   40     CONTINUE
          WRITE (IU'4) ((ML(II,IJ),IJ=1,NI),II=1,NI)
C         CALL USWFM (10HMATRIX NL:,10,NL,10,NS,NI,4)      !**
C         CALL USWFM (10HMATRIX ML:,10,ML,10,NI,NI,4)      !**
          IF (I.GE.NS) GO TO 900
   50     I=I+1
          IF (I.GT.NS) GO TO 900
          WRITE (5,15)
   15     FORMAT (1X,16HNEXT EIGENVALUE:)
          GO TO 999
C
C
  100     CONTINUE
C********************COMPLEX NULL SPACE FORMULATION ***********
C********************FORM S-LAMBDA-C, NSX(2NS+NI) *************
          DO 110 II=1,NS
          DO 110 IJ=1,NS
          S(II,IJ)=-A(II,IJ)
          IF (II.EQ.IJ) S(II,IJ)=S(II,IJ)+LRE(I)
  110     CONTINUE
        INS=NS+1
        NS2=2*NS
        NI2=2*NI
          DO 120 II=1,NS
          DO 120 IJ=INS,NS2
          S(II,IJ)=0.0
          IJDUM=IJ-NS
```

```
            IF (II.EQ.IJDUM) S(II,IJ)=LIM(I)
  120       CONTINUE
           IINS=NS2+1
           ILC=NS2+NI
            DO 130 II=1,NS
            DO 130 IJ=IINS,ILC
            IJDUM=IJ-NS2
            S(II,IJ)=B(II,IJDUM)
  130       CONTINUE
C           CALL USWFM (11HMATRIX SLC:,11,S,10,NS,ILC,4)  !**
C*************************** CALL NSA ***************************
C           WRITE (5,4)                          !**
C   4       FORMAT (1X,'NULL SPACE OF SLC, X=KLC ')          !**
            CALL NSA (NS,ILC,S,10,30,X,30,20,ZERO,IDGT,SCOPY,SPP,SP)
C****************** PARTITION X=KLC INTO NLC,PLC,AND MLC ******
           IS=NS+NI
            DO 140 II=1,NS
            DO 140 IJ=1,IS
            NLC(II,IJ)=X(II,IJ)
  140       CONTINUE
            WRITE (IU'3) ((NLC(II,IJ),IJ=1,IS),II=1,NS)
C
            DO 150 II=INS,NS2
            DO 150 IJ=1,IS
           IIDUM=II-NS
            PLC(IIDUM,IJ)=X(II,IJ)
  150       CONTINUE
            WRITE (IU'4) ((PLC(II,IJ),IJ=1,IS),II=1,NS)
C
            DO 160 II=IINS,ILC
            DO 160 IJ=1,IS
           IJDUM=II-NS2
            MLC(IJDUM,IJ)=X(II,IJ)
  160       CONTINUE
            WRITE (IU'5) ((MLC(II,IJ),IJ=1,IS),II=1,NI)
C
C           CALL USWFM (11HMATRIX NLC:,11,NLC,10,NS,IS,4)    !**
C           CALL USWFM (11HMATRIX PLC:,11,PLC,10,NS,IS,4)    !**
C           CALL USWFM (11HMATRIX MLC:,11,MLC,10,NI,IS,4)    !**
            IF (NS.EQ.NI) GO TO 215
C****************** FORM ALPHA,TRANSPOSE ******************
            DO 170 II=1,NS2
            DO 170 IJ=1,IS
            ALPHA(II,IJ)=X(II,IJ)
            IF (II.GT.NS) ALPHA(II,IJ)=-X(II,IJ)
  170       CONTINUE
C           CALL USWFM (14HMATRIX ALPHAT:,14,ALPHA,20,NS2,IS,4) !**
            CALL TRANS (ALPHA,NS2,IS)
C           CALL USWFM (20HTRANSPOSE OF ALPHAT:,20,ALPHA,20,IS,NS2,4) !*
C******************** CALL NSA ********************************
C           WRITE (5,5)                          !**
C   5       FORMAT (1X,'NULL SPACE OF ALPHA, KA ')           !**
            CALL NSA (IS,NS2,ALPHA,20,20,KA,20,10,ZERO,IDGT,ACOPY,APP,AP)
           NMI=NS-NI
C           CALL USWFM (10HMATRIX KA:,10,KA,20,NS2,NMI,4)   !**
C****************FORM BETA, TRANSPOSE **********************
```

```
          DO 180 II=1,NS
          DO 180 IJ=1,IS
          BETA(II,IJ)=PLC(II,IJ)
   180    CONTINUE
C
          DO 190 II=INS,NS2
          DO 190 IJ=1,IS
         IDUM=II-NS
          BETA(II,IJ)=NLC(IDUM,IJ)
   190    CONTINUE
C         CALL USWFM (13HMATRIX BETAT:,13,BETA,20,NS2,IS,4) !**
          CALL TRANS (BETA,NS2,IS)
C         CALL USWFM (19HTRANSPOSE OF BETAT:,19,BETA,20,IS,NS2,4)!**
C****************** CALL NSA **********************************
C         WRITE (5,6)                            !**
C    6    FORMAT (1X,'NULL SPACE OF BETA, KB ')        !**
          CALL NSA (IS,NS2,BETA,20,20,KB,20,10,ZERO,IDGT,ACOPY,APP,AP)
C         CALL USWFM (10HMATRIX KB:,10,KB,20,NS2,NMI,4)  !**
C************************ FORM GAMA, TRANSPOSE *****************
          DO 200 II=1,NS2
          DO 200 IJ=1,NMI
          GAMA(II,IJ)=KA(II,IJ)
   200    CONTINUE
C
          NMI2=2*NMI
          NMI1=NMI+1
          DO 210 II=1,NS2
          DO 210 IJ=NMI1,NMI2
         NMIDUM=IJ-NMI
          GAMA(II,IJ)=KB(II,NMIDUM)
   210    CONTINUE
C
C         CALL USWFM (13HMATRIX GAMAT:,13,GAMA,20,NS2,NMI2,4)  !**
          CALL TRANS (GAMA,NS2,NMI2)
C         CALL USWFM (19HTRANSPOSE OF GAMAT:,19,GAMA,20,NMI2,NS2,4)!*
C********************** CALL NSA *******************************
C         WRITE (5,7)
C   7    FORMAT (1X,'NULL SPACE OF GAMA, STAR ')      !***
          CALL NSA (NMI2,NS2,GAMA,20,20,STAR,20,20,ZERO,IDGT,ACOPY,APP,AP)
          GO TO 216
   215    DO 216 II=1,NS2
          DO 216 IJ=1,NI2
          STAR(II,IJ)=FLOAT(0)
          IF (II.EQ.IJ) STAR(II,IJ)=FLOAT(1)
   216    CONTINUE
C         CALL USWFM (12HMATRIX STAR:,12,STAR,20,NS2,NI2,4)  !**
C*********************** PARTITION STAR *********************
          DO 220 II=1,NS
          DO 220 IJ=1,NI2
          QL(II,IJ)=STAR(II,IJ)
   220    CONTINUE
          WRITE (IU'6) ((QL(II,IJ),IJ=1,NI2),II=1,NS)
C
          DO 230 II=INS,NS2
          DO 230 IJ=1,NI2
         IDUM=II-NS
```

```
            RL(IDUM,IJ)=STAR(II,IJ)
  230       CONTINUE
            WRITE (IU'7) ((RL(II,IJ),IJ=1,NI2),II=1,NS)
C           CALL USWFM (10HMATRIX QL:,10,QL,10,NS,NI2,4)  !**
C           CALL USWFM (10HMATRIX RL:,10,RL,10,NS,NI2,4)  !**
C
C*******************SET THE CONJUGATE VALUES ******************
            IC=I+1
            IRS=202
            IU=IC+20
          OPEN (ACCESS='RANDOM',RECORD SIZE=IRS
         1,UNIT=IU,MODE='BINARY',DEVICE='DSK',DISPOSE='SAVE')
            LRE(IC)=LRE(I)
            LIM(IC)=-LIM(I)
            WRITE (IU'1) LRE(IC),LIM(IC)
            WRITE (5,22) IC,LRE(IC),LIM(IC)
  22       FORMAT (1X,6HLAMBDA,I2,6H:REAL=,E15.6,2X,6H,IMAG=,E15.6)
            IF (IC.GE.NS) GO TO 900
            I=I+2
            WRITE (5,15)
            GO TO 999
C
  900      WRITE (5,8)
    8      FORMAT (1X,29HWISH TO EXIT FROM THIS MODE? )
            READ (5,*) KK
            IF (KK.LE.0) GO TO 910
            WRITE (5,9)
    9      FORMAT (1X,27(1H*),18H  EXITING MODE 2  ,25(1H*))
            RETURN
            END
```

```
c*****************************************************************
c******************************************************************
          SUBROUTINE NSA(M,N,S,iis,ijs,X,iix,ijx,ZERO,IDGT,SCOPY,SPP,SP)
C-Function: Calculates a basis for the Null Space of a MxN matrix S.
C-IMSL routines called: UERSET,UERTST,LEQT2F,VSRTU,VSRTR,(USWFM).
C-Spectral Assignment routines: -
C-Logical devices; Input Unit:  (5)   Output Unit:  (5)
C              Storage Unit(s): -
C-Random Access Files: -
         real s(iis,ijs),scopy(iis,ijs),spp(iis,ijX),sp(iis,iIs)
          real x(iix,ijx),fac,wk1(10),wk2(132),wk3(30)
        INTEGER M,N,IM,IN,JN,PV(30),IPV(30),K,L,DUM
        INTEGER PVCOPY(30),RPV(30)
          DO 90 I=1,M
          DO 90 J=1,N
          SCOPY(I,J)=S(I,J)
   90     CONTINUE
C         WRITE (5,2) ZERO,IDGT                        !**
C    2    FORMAT (1X,5HZERO=,F15.12,1X,5HIDGT=,I2)  !**
        DUM=N-M
        IN=N
        JN=N
        IM=1
          DO 20 I=1,N
          PV(I)=I
   20     CONTINUE
        IF (ABS(S(IM,IN)).GT.ABS(ZERO)) GO TO 30
   70   IN=IN-1
          GO TO 20
   30     IK=PV(JN)
        PV(JN)=PV(IN)
        IF (IN.EQ.JN) GO TO 50
        PV(IN)=IK
C***************EXCHANGE COLUMNS IN AND JN******************
          DO 40 I=1,N
          IPV(I)=I
   40     CONTINUE
        K=IPV(IN)
        IPV(IN)=IPV(JN)
        IPV(JN)=K
          CALL VSRTU (S,IIS,M,N,0,IPV,WK1)
        IN=JN
   50   CONTINUE
        IF (IM.EQ.M) GO TO 80
        L=IM+1
C*************************GAUSSIAN PROCESS*******************
          DO 60 IL=L,N
          IF (ABS(S(IL,IN)).LE.ABS(ZERO)) GO TO 60
          FAC=S(IL,IN)/S(IM,IN)
          DO 60 I=1,N
          S(IL,I)=S(IL,I)-FAC*S(IM,I)
   60     CONTINUE
        JN=JN-1
        IM=IM+1
          GO TO 70
   80   CONTINUE
```

```
C         CALL USWFM (13HS TRIANGULAR:,13,S,IIS,M,N,4)  !**
C         WRITE (5,4)                                  !**
C    4    FORMAT(1X ,'PERMUTATION VECTOR :')          !**
C         DO 81 I=1,N                                  !**
C         WRITE (5,*) PV(I)                            !**
C   81    CONTINUE                                     !**
C         CALL USWFM (10HMATRIX S :,10,SCOPY,IIS,M,N,4)  !**
C************SHUFFLE SCOPY,USING PVCOPY**********************
          DO 120 I=1,N
          PVCOPY(I)=PV(I)
  120     CONTINUE
          CALL VSRTU (SCOPY,IIS,M,N,0,PVCOPY,WK1)
C*******************SCOPY NOW CONTAINS SBAR*******************
C         CALL USWFM (12HMATRIX SBAR:,12,SCOPY,IIS,M,N,4)  !**
C*******************PARTITION SBAR***************************
          DO 100 I=1,M
          DO 100 J=1,DUM
          SPP(I,J)=SCOPY(I,J)
  100     CONTINUE
C         CALL USWFM (11HMATRIX SPP:,11,SPP,IIS,M,DUM,4)!**
          DO 110 I=1,M
          DO 110 J=1,M
          JDUM=J+DUM
          SP(I,J)=SCOPY(I,JDUM)
  110     CONTINUE
C         CALL USWFM (10HMATRIX SP:,10,SP,IIS,M,M,4) !***
C*************LINEAR EQUATION SOLUTION***********************
          IT=IDGT
          CALL LEQT2F (SP,DUM,M,IIS,SPP,IT,WK2,IER)
          CALL UERSET (3,LEVOLD)
          CALL UERTST (IER,6HLEQT2F)
C         WRITE (5,3) IT                                   !**
C    3    FORMAT(1X ,31HIDGT ON RETURN FROM LEQT2F IS =,I3)  !**
C************************SPP CONTAINS XP********************
C***********SORT PV*****************************************
          DO 130 I=1,N
          RPV(I)=I
  130     CONTINUE
          CALL VSRTR (PV,N,RPV)
C************************FORM X******************************
          DO 140 I=1,DUM
          DO 140 J=1,DUM
          X(I,J)=FLOAT(0)
          IF (I.EQ.J) X(I,J)=-FLOAT(1)
  140     CONTINUE
          IIDUM=DUM+1
          DO 150 I=IIDUM,N
          DO 150 J=1,DUM
          IDUM=I-DUM
          X(I,J)=SPP(IDUM,J)
  150     CONTINUE
C**********************SHUFFLE ROWS OF X*********************
C         CALL USWFM (20HX BEFORE SHUFFLING :,20,X,IIX,N,DUM,4)!**
          CALL VSRTU (X,IIX,N,DUM,1,RPV,WK3)
C         CALL USWFM (20HBASIS VECTORS ARE  :,20,X,IIX,N,DUM,4)!**
           RETURN

        END
```

```
C*****************************************************************
C*****************************************************************
      SUBROUTINE MODEO
C-Function: Signature.
C-IMSL routines called: -
C-Spectral Assignment routines: -
C-Logical devices; Input Unit:   -   Output Unit:   5
C             Storage Unit(s): -
C-Random Access Files: -
      WRITE (5,1)
    1 FORMAT (/,13X,3H***,/,13X,3H***,16X,23HOld Dominion University,/
     1,13X,3H***,10X,36HDepartment of Electrical Engineering,/,13x,3H***
     2,21X,14HMohsen Marefat,/,13X,3H***,21X,14HSeptember 1982)
      WRITE (5,2)
    2 FORMAT (4X,3H***,3X,2H**,1X,3H***,3X,3H***,/,2X,14(1H*),3X,3H***,
     1/,1X,4(6H***   ),3X,31HThe Spectral Assignment Package,/,1X
     2,4(6H***   ),3X,31(1H=),/,2X,19(1H*),/,1X,3(6H   ***))
      WRITE (5,3)
    3 FORMAT (//,6X,52HDocumentation and a user guide for this CAD progr
     1am ,/,6x,34Hpackage is avialable upon request.,/,
     26X,45HContact Dr. R.R. Mielke at the EE department.)
      RETURN
      END
```

```
C******************************************************************
C******************************************************************
        SUBROUTINE TRANS (A,IM,IN)
C-Function: Returns the transpose of matrix [A] in A.
C-IMSL routines called: -
C-Spectral Assignment routines: -
C-Logical devices; Input Unit:   -   Output Unit:   -
C              Storage Unit(s): -
C-Random Access Files: -
        REAL A(20,20),AT(20,20)
        DO 10 I=1,IM
        DO 10 J=1,IN
        AT(J,I)=A(I,J)
 10     CONTINUE
        DO 20 I=1,IN
        DO 20 J=1,IM
        A(I,J)=AT(I,J)
 20     CONTINUE
        RETURN
        END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C*************************************************************
C*************************************************************
        SUBROUTINE MODE3
C-Function: Main routine for Eigenvector Assignment.
C-IMSL routines called: UERTST,UERSET,USWFV,USWFM,LLSQF,VMULFF.
C-Spectral Assignment routines: GAIN,IMP,PROJ,NORM.
C-Logical devices; Input Unit:   5   Output Unit:   5
C              Storage Unit(s): IU=20,IUT=20+NS+1,IU=20+J for J=1,NS.
C-Random Access Files: SYSTEM.DAT,CURRNT.DAT,FORxx.DAT where xx=20+J
C                        for J=1,NS.
C NULL SPACE ARRAYS
        REAL ML(10,10),NL(10,10)
        REAL NLC(10,20),PLC(10,20),MLC(10,20)
        REAL STAR(20,20),QL(10,20),RL(10,20)
C AUX. ARRAYS
        REAL WKAREA(130),CP(20,20),ATA(20,20),ATAI(20,20)
        REAL PNL(10,10),PSTAR(20,20),XX(10,10)
        REAL LRE(10),LIM(10)
C MODE 3 ARRAYS
      REAL VRE(10,10),VIM(10,10),VD(20),VA(20),E(20),X(20),H(20)
      REAL WJ(10),W(10,10),V(10,10),VINV(10,10),F(10,10),AHAT(10,10)
        INTEGER IP(10)
        REAL A(10,10),B(10,10),C(10,10)
        COMMON/SYS/A,B,C,ZERO,IDGT,NS,NI,NO
        COMMON/AUG/F,AHAT/EIG/LRE,LIM
        COMMON/VEC/VA,E,X,WJ,W,XX,V,VINV
        COMMON/NSPA/ML,NL,NLC,PLC,MLC,STAR,QL,RL
C****************** READ SYSTEM DATA **************************
        CALL UERSET (3,LEVOLD)
        IRS=102
        IU=20
        READ (IU'1) NS,NI,NO,IDGT,ZERO
        READ (IU'2) ((A(I,J),J=1,NS),I=1,NS)
        READ (IU'3) ((B(I,J),J=1,NI),I=1,NS)
C
        IUT=IU+NS+1
      OPEN (FILE='CURRNT.DAT',ACCESS='RANDOM',RECORD SIZE=IRS
     1,UNIT=IUT,MODE='BINARY',DEVICE='DSK',DISPOSE='SAVE')
        WRITE (5,1)
  1     FORMAT (1X,20(1H*),31H MODE 3:EIGENVECTOR ASSIGNMENT ,19(1H*)
     1,//,1X,10(1H*),30H ENTER OR CHANGE EIGENVECTORS:,//)
        WRITE (5,11)
 11     FORMAT (1X,16HPREVIOUS VALUES?)
        READ (5,*) KO
        IF (KO.LE.0) GO TO 910
        IFLAG=1
        READ (IUT'1) ((V(II,IJ),IJ=1,NS),II=1,NS)
        READ (IUT'2) ((XX(II,IJ),IJ=1,NS),II=1,NI)
 910    J=1
 999    CONTINUE
        IU=J+20
        IRS=202
        OPEN (ACCESS='RANDCM',RECORD SIZE=IRS,UNIT=IU
     1,MODE='BINARY',DEVICE='DSK',DISPOSE='SAVE')
        IF (IFLAG.NE.1) GO TO 13
        READ (IU'2) ((VRE(IV,J),VIM(IV,J)),IV=1,NS)
```

```
   13 WRITE (5,14) J
   14 FORMAT (1X,13HEIGENVECTOR V,I2,1H:,3X,6H(REAL),14X,6H(IMAG))
      DO 10 IV=1,NS
      WRITE (5,15) VRE(IV,J),VIM(IV,J)
   15 FORMAT (15X,E15.6,5X,E15.6)
   10 CONTINUE
      WRITE (5,16)
   16 FORMAT (1X,15HWISH TO CHANGE?)
      READ (5,*) K1
      IF (K1.LE.0) GO TO 50
      WRITE (5,17)
   17 FORMAT (1X,28HENTER A NEW DESIRED VECTOR :)
      READ (5,*) ((VRE(IV,J),VIM(IV,J)),IV=1,NS)
      WRITE (IU'2) ((VRE(IV,J),VIM(IV,J)),IV=1,NS)
C***************** IS V-J REAL OR COMPLEX ? ********************
      READ (IU'1) LRE(J),LIM(J)
      IF (ABS(LIM(J)).GT.ABS(ZERO)) GO TO 100
C******************* REAL EIGENVECTOR PROJECTION ****************
      READ (IU'3) ((NL(II,IJ),IJ=1,NI),II=1,NS)
C     CALL USWFM (10HMATRIX NL:,10,NL,10,NS,NI,4)     !**
      CALL PROJ (NL,NS,NI,10,10,PNL,CP,ATA,ATAI,IDGT)
C     CALL USWFM (11HMATRIX PNL:,11,PNL,10,NS,NS,4)  !**
C******************* PROJECT VD ONTO COLUMN SPACE OF N-LAMBDA **
   25 DO 30 IV=1,NS
      VD(IV)=VRE(IV,J)
   30 CONTINUE
      CALL USWFV(15HDESIRED VECTOR:,15,VD,NS,1,4)
      CALL VMULFF (PNL,VD,NS,NS,1,10,20,VA,20,IER)
C     CALL UERTST (IER,6HVMULFF)
      CALL USWFV (15HACTUAL VECTORT:,15,VA,NS,1,4)
C     CALL USWFM (15HVA FROM USWFM :,15,VA,20,NS,1,4)!**
C******************* FIND THE ERROR VECTOR *********************
      CALL IMP(PNL,NS,10)
C     CALL USWFM (13HMATRIX I-PNL:,13,PNL,10,NS,NS,4)!**
      CALL VMULFF (PNL,VD,NS,NS,1,10,20,E,20,IER)
C     CALL UERTST (IER,6HVMULFF)
      CALL USWFV (14HERROR VECTORT:,14,E,NS,1,4)
      CALL NORM (VD,NS,XVD)
      CALL NORM (VA,NS,XVA)
      CALL NORM (E,NS,XE)
      WRITE (5,18) XVD,XVA,XE
   18 FORMAT (1X,31HLENGTH OF THE DESIRED VECTOR  =,F15.6,/
     1,1X,31HLENGTH OF THE PROJECTED VECTOR=,F15.6,/
     2,1X,31HLENGTH OF THE ERROR VECTOR    =,F15.6)
      WRITE (5,21)
   21 FORMAT (1X,24HIS THE ERROR ACCEPTABLE?)
      READ (5,*) KK
      IF (KK.GT.0) GO TO 45
      WRITE (5,17)
      READ (5,*) ((VRE(IV,J),VIM(IV,J)),IV=1,NS)
      WRITE (IU'2) ((VRE(IV,J),VIM(IV,J)),IV=1,NS)
      CALL IMP(PNL,NS,10)
      GO TO 25
   45 DO 46 IV=1,NS
      V(IV,J)=VA(IV)
   46 CONTINUE
```

```
C***************** SOLVE NL*X=VA FOR X **************************
C NOTE: VA IS DESTROYED!
           CALL LLSQF (NL,10,NS,NI,VA,-1.0,NI,X,H,IP,IER)
           CALL UERTST (IER,6HLLSQF )
C          CALL USWFV (10HVECTOR XT:,10,X,NI,1,4)          !**
           DO 49 IV=1,NI
           XX(IV,J)=X(IV)
   49      CONTINUE
           IF (J.GE.NS) GO TO 900
   50      J=J+1
           IF (J.GT.NS) GO TO 900
           WRITE (5,19)
   19      FORMAT (1X,17HNEXT EIGENVECTOR:)
           GO TO 999
C
  100      CONTINUE
C****************** COMPLEX EIGENVECTOR ASSIGNMENT **************
           IS=NS+NI
           NI2=2*NI
           NS2=2*NS
           INS=NS+1
           READ (IU'6) ((QL(II,IJ),IJ=1,NI2),II=1,NS)
           READ (IU'7) ((RL(II,IJ),IJ=1,NI2),II=1,NS)
C          CALL USWFM (10HMATRIX QL:,10,QL,10,NS,NI2,4)   !**
C          CALL USWFM (10HMATRIX RL:,10,RL,10,NS,NI2,4)   !**
C*************** FORM STAR AND FIND P-STAR *******************
           DO 105 II=1,NS
           DO 105 IJ=1,NI2
           STAR(II,IJ)=QL(II,IJ)
  105      CONTINUE
           DO 110 II=INS,NS2
           DO 110 IJ=1,NI2
           IDUM=II-NS
           STAR(II,IJ)=RL(IDUM,IJ)
  110      CONTINUE
           CALL PROJ (STAR,NS2,NI2,20,20,PSTAR,CP,ATA,ATAI,IDGT)
C          CALL USWFM (12HMATRIX STAR:,12,STAR,20,NS2,NI2,4) !**
C          CALL USWFM (13HMATRIX PSTAR:,13,PSTAR,20,NS2,NS2,4)!**
C************* PROJECT VD CNTO THE COLUMN SPACE OF STAR ********
  114      DO 115 IV=1,NS
           VD(IV)=VRE(IV,J)
  115      CONTINUE
           DO 120 IV=INS,NS2
           IVDUM=IV-NS
           VD(IV)=VIM(IVDUM,J)
  120      CONTINUE
           CALL USWFV (11HCOMPLEX VD:,11,VD,NS2,1,4)
           CALL VMULFF (PSTAR,VD,NS2,NS2,1,20,20,VA,20,IER)
C          CALL UERTST (IER,6HVMULFF)
           CALL USWFV (15HACTUAL VECTORT:,15,VA,NS2,1,4)
C***************** FIND THE ERROR VECTOR **********************
           CALL IMP(PSTAR,NS2,20)
C          CALL USWFM (15HMATRIX I-PSTAR:,15,PSTAR,20,NS2,NS2,4) !**
           CALL VMULFF (PSTAR,VD,NS2,NS2,1,20,20,E,20,IER)
C          CALL UERTST (IER,6HVMULFF)
           CALL USWFV (14HERROR VECTORT:,14,E,NS2,1,4)
```

```
             CALL NORM (VD,NS2,XVD)
             CALL NORM (VA,NS2,XVA)
             CALL NORM (E,NS2,XE)
             WRITE (5,18) XVD,XVA,XE
             WRITE (5,21)
             READ (5,*) KM
             IF (KM.GT.0) GO TO 134
             WRITE (5,17)
             READ (5,*) ((VRE(IV,J),VIM(IV,J)),IV=1,NS)
             WRITE (IU'2) ((VRE(IV,J),VIM(IV,J)),IV=1,NS)
             CALL IMP(PSTAR,NS2,20)
             GO TO 114
  134        IC=J+1
             DO 136 IV=1,NS
             V(IV,J)=VA(IV)
             IVNS=IV+NS
             V(IV,IC)=VA(IVNS)
  136        CONTINUE
             CALL LLSQF (STAR,20,NS2,NI2,VA,-1.0,NI2,X,H,IP,IER)
             CALL UERTST (IER,6HLLSQF )
C            CALL USWFV (16H[XX]-j,[XX]-J+1:,16,X,NI2,1,4)     !**
             DO 137 IV=1,NI
             XX(IV,J)=X(IV)
             IVNS=IV+NI
             XX(IV,IC)=X(IVNS)
  137        CONTINUE
C            CALL USWFM (13HMATRIX  [XX]:,13,XX,10,NI,NS,4)      !**
C******************** SET THE CONJUGATE VALUES ****************
             IRS=202
             IU=IC+20
          OPEN (ACCESS='RANDOM',RECORD SIZE=IRS
         1,UNIT=IU,MODE='BINARY',DEVICE='DSK',DISPOSE='SAVE')
             DO 220 IV=1,NS
             VRE(IV,IC)=VRE(IV,J)
             VIM(IV,IC)=-VIM(IV,J)
  220        CONTINUE
             WRITE (IU'2) ((VRE(IV,IC),VIM(IV,IC)),IV=1,NS)
             WRITE (5,14) IC
             DO 230 IV=1,NS
             WRITE (5,15) VRE(IV,IC),VIM(IV,IC)
  230        CONTINUE
             IF (IC.GE.NS) GO TO 900
             J=J+1
             GO TO 50
  900        CONTINUE
             WRITE (5,901)
  901        FORMAT(1X,49H==========CONTENTS OF "CURRNT" DATA FILE INCLUDE:)
C            CALL USWFM (13HMATRIX  [XX]:,13,XX,10,NI,NS,4)       !**
             CALL USWFM (10HMATRIX V :,10,V,10,NS,NS,4)        !**
             WRITE (5,902)
  902        FORMAT(1X,44HWISH TO DISPLAY THE NORMALIZED EIGENVECTORS?)
             READ (5,*) KS
             IF (KS.LE.0) GO TO 903
             CALL DSPLAY(NS,ZERO)
  903        CALL GAIN
             CALL USWFM (14HGAIN MATRIX F:,14,F,10,NI,NS,4)   !**
```

```
          CALL USWFM (12HMATRIX AHAT:,12,AHAT,10,NS,NS,4)    !**
          WRITE (5,8)
      8   FORMAT (1X,29HWISH TO EXIT FROM THIS MODE? )
          READ (5,*) KT
          IF (KT.GT.0) GO TO 920
          IFLAG=1
          GO TO 910
    920   WRITE (IUT'1) ((V(II,IJ),IJ=1,NS),II=1,NS)
          WRITE (IUT'2) ((XX(II,IJ),IJ=1,NS),II=1,NI)
          WRITE (IUT'3) ((W(II,IJ),IJ=1,NS),II=1,NI)
          WRITE (IUT'4) ((F(II,IJ),IJ=1,NS),II=1,NI)
          WRITE (IUT'5) ((AHAT(II,IJ),IJ=1,NS),II=1,NS)
          WRITE (5,9)
      9  FORMAT (1X,27(1H*),18H  EXITING MODE 3  ,25(1H*))
          RETURN
          END
```

```
c***************************************************************
c****************************************************************
         SUBROUTINE PROJ(A,M,N,IM,IN,P,CP,ATA,ATAI,IDGT)
C-Function: Calculates a projection matrix [P] for the allowable
C-             space represented by [A].
C-IMSL routines called: UERSET,UERTST,LINV2F,VMULFF,VMULFM,VMULFP,
C                         (USWFM).
C-Spectral Assignment routines: -
C-Logical devices; Input Unit:   -   Output Unit:  (5)
C              Storage Unit(s): -
C-Random Access Files: -
         REAL A(IM,IN),ATA(N,N),ATAI(N,N)
         REAL P(IM,IM),CP(N,M),WKAREA(460)
         CALL UERSET (3,LEVOLD)
C        CALL USWFM(9HMATRIX A:,9,A,IM,M,N,4)  !**
         DO 10 I=1,N
         DO 10 J=1,N
         ATAI(I,J)=FLOAT(0)
         IF (I.EQ.J) ATAI(I,J)=FLOAT(1)
  10     CONTINUE
         CALL VMULFM (A,A,M,N,N,IM,IM,ATA,N,IER)
C        CALL UERTST (IER,6HVMULFM)
C        CALL USWFM (11HMATRIX ATA:,11,ATA,N,N,N,4)     !**
         CALL LINV2F (ATA,N,N,ATAI,IDGT,WKAREA,IER)
         CALL UERTST (IER,6HLINV2F)
C        CALL USWFM (12HMATRIX ATAI:,12,ATAI,N,N,N,4)  !**
         CALL VMULFP (ATAI,A,N,N,M,N,IM,CP,N,IER)
         CALL UERTST (IER,6HVMULFP)
C        CALL USWFM (10HMATRIX CP:,10,CP,N,N,M,4)       !**
         CALL VMULFF (A,CP,M,N,M,IM,N,P,IM,IER)
         CALL UERTST (IER,6HVMULFF)
C        CALL USWFM (10HMATRIX P :,10,P,IM,M,M,4)       !**
         RETURN
         END
```

```
C****************************************************************
C****************************************************************
        SUBROUTINE NORM(V,N,XNORM)
C-Function: Calculates the norm of an N-vector V.
C-IMSL routines called: -
C-Spectral Assignment routines: -
C-Logical devices; Input Unit:   -   Output Unit:   -
C             Storage Unit(s): -
C-Random Access Files: -
        REAL V(N)
        XNORM=FLOAT(0)
        DO 10 I=1,N
        XNORM=XNORM+V(I)**2
 10     CONTINUE
        XNORM=SQRT(XNORM)
        RETURN
        END
```

```
C****************************************************************
C****************************************************************
         SUBROUTINE IMP(P,N,IN)
C-Function: Returns [P]=[I]-[P].
C-IMSL routines called: -
C-Spectral Assignment routines: -
C-Logical devices; Input Unit:   -   Output Unit:   -
C              Storage Unit(s): -
C-Random Access Files: -
         REAL P(IN,IN)
         DO 10 I=1,N
         DO 10 J=1,N
         P(I,J)=-P(I,J)
         IF (I.EQ.J) P(I,J)=P(I,J)+FLOAT(1)
  10     CONTINUE
         return
         end
```

```
C*****************************************************************
C*****************************************************************
            SUBROUTINE GAIN
C-Function: Calculates the Gain matrix,[F].
C-IMSL routines called: UERSET,UERTST,LINV2F,LLSQF,VMULFF,(USWFM,USWFV).
C-Spectral Assignment routines: -
C-Logical devices; Input Unit:   -   Output Unit:  (5)
C                Storage Unit(s): IU=20+J for J=1,NS.
C-Random Access Files: FORxx.DAT where xx=20+J for J=1,NS.
C NULL SPACE ARRAYS
            REAL ML(10,10),NL(10,10)
            REAL NLC(10,20),PLC(10,20),MLC(10,20)
            REAL STAR(20,20),QL(10,20),RL(10,20)
C AUX. ARRAYS
            REAL WKAREA(130),H(20)
C MODE 3 ARRAYS
         REAL XX(10,10),VA(20),E(20),X(20),LRE(10),LIM(10),WJ(10)
         REAL W(10,10),V(10,10),VINV(10,10),F(10,10),AHAT(10,10)
            INTEGER IP(10)
            REAL A(10,10),B(10,10),C(10,10)
            COMMON/SYS/A,B,C,ZERO,IDGT,NS,NI,NO
            COMMON/AUG/F,AHAT/EIG/LRE,LIM
            COMMON/VEC/VA,E,X,WJ,W,XX,V,VINV
            COMMON/NSPA/ML,NL,NLC,PLC,MLC,STAR,QL,RL
            CALL UERSET (3,LEVOLD)
C           WRITE (5,1)
C   1       FORMAT (1X,'SUBROUTINE GAIN ++++++++++++++++++++++++')
            IRS=202
            J=1
   10       IU=J+20
            OPEN (ACCESS='RANDOM',RECORD SIZE=IRS,UNIT=IU
        1,MODE='BINARY',DEVICE='DSK',DISPOSE='SAVE')
C*************** Is Lambda-J real? ******************************
            READ (IU'1) LRE(J),LIM(J)
            IF (ABS(LIM(J)).GT.ABS(ZERO)) GO TO 30
C*************** Find real WJ=J-th column of [W] ****************
            READ (IU'4) ((ML(II,IJ),IJ=1,NI),II=1,NI)
C           CALL USWFM (10HMATRIX ML:,10,ML,10,NI,NI,4)     !**
            DO 20 IV=1,NI
            X(IV)=XX(IV,J)
   20       CONTINUE
C********** FORM WJ=[ML]*X AND PUT WJ IN J-TH COLUMN OF [W] ***
            CALL VMULFF (ML,X,NI,NI,1,10,20,WJ,10,IER)
            CALL UERTST (IER,6HVMULFF)
C           CALL USWFV (10HVECTOR WJ:,10,WJ,NI,1,4)     !**
            DO 25 IV=1,NI
            W(IV,J)=WJ(IV)
   25       CONTINUE
   29       IF (J.GE.NS) GO TO 100
            J=J+1
            GO TO 10
C*************** Find complex WJ's ******************************
   30       IS=NS+NI
            NI2=2*NI
            NS2=2*NS
            INS=NS+1
```

```
            READ (IU'3) ((NLC(II,IJ),IJ=1,IS),II=1,NS)
            READ (IU'4) ((PLC(II,IJ),IJ=1,IS),II=1,NS)
            READ (IU'5) ((MLC(II,IJ),IJ=1,IS),II=1,NI)
C           CALL USWFM (11HMATRIX NLC:,11,NLC,10,NS,IS,4)      !**
C           CALL USWFM (11HMATRIX PLC:,11,PLC,10,NS,IS,4)      !**
C           CALL USWFM (11HMATRIX MLC:,11,MLC,10,NI,IS,4)      !**
            IC=J+1
C****** FORM ALPHAT AND SOLVE [ALPHAT]*X=VA FOR X *************
            DO 135 II=1,NS
            DO 135 IJ=1,IS
            STAR(II,IJ)=NLC(II,IJ)
  135       CONTINUE
            DO 140 II=INS,NS2
            DO 140 IJ=1,IS
            IDUM=II-NS
            STAR(II,IJ)=-PLC(IDUM,IJ)
  140       CONTINUE
C           CALL USWFM (14HMATRIX ALPHAT:,14,STAR,20,NS2,IS,4) !**
            DO 40 IV=1,NS
            VA(IV)=V(IV,J)
            E(IV)=VA(IV)
   40       CONTINUE
            DO 50 IV=INS,NS2
            IVDUM=IV-NS
            VA(IV)=V(IVDUM,IC)
            E(IV)=VA(IV)
   50       CONTINUE
            CALL LLSQF (STAR,20,NS2,IS,VA,-1.0,IS,X,H,IP,IER)
            CALL UERTST (IER,6HLLSQF )
C           CALL USWFV (10HVECTOR XT:,10,X,IS,1,4)                !**
C******** FORM WJ=[MLC]*XC AND PUT WJ IN THE J-TH COLUMN OF [W] *
            CALL VMULFF (MLC,X,NI,IS,1,10,20,WJ,10,IER)
            CALL UERTST (IER,6HVMULFF)
C           CALL USWFV (10HVECTOR WJ:,10,WJ,NI,1,4)              !**
            DO 60 IV=1,NI
            W(IV,J)=WJ(IV)
   60       CONTINUE
C
            IF (J.EQ.IC) GO TO 29
            J=IC
C********* FORM BETAT AND SOLVE [BETAT]*X=E(=VA) FOR X ********
            DO 180 IV=1,IS
            X(IV)=FLOAT(0)
  180       CONTINUE
            DO 185 II=1,NS
            DO 185 IJ=1,IS
            STAR(II,IJ)=PLC(II,IJ)
  185       CONTINUE
            DO 190 II=INS,NS2
            DO 190 IJ=1,IS
            IDUM=II-NS
            STAR(II,IJ)=NLC(IDUM,IJ)
  190       CONTINUE
C            CALL USWFM (13HMATRIX BETAT:,13,STAR,20,NS2,IS,4)   !**
            DO 70 IV=1,NS2
            VA(IV)=E(IV)
```

```
  70    CONTINUE
        GO TO 50
C************** Print [W],-[W], then find [F], and [AHAT] *******
 100    CONTINUE
C       CALL USWFM (11HMATRIX [W]:,11,W,10,NI,NS,4)          !**
        DO 80 II=1,NI
        DO 80 IJ=1,NS
        W(II,IJ)=-W(II,IJ)
  80    CONTINUE
C       CALL USWFM (12HMATRIX -[W]:,12,W,10,NI,NS,4)       !**
C       CALL USWFM (10HMATRIX V :,10,V,10,NS,NS,4)        !**
        CALL LINV2F (V,NS,10,VINV,IDGT,WKAREA,IER)
        CALL UERTST (IER,6HLINV2F)
C       CALL USWFM (12HMATRIX VINV:,12,VINV,10,NS,NS,4)   !**
        CALL VMULFF (W,VINV,NI,NS,NS,10,10,F,10,IER)
        CALL UERTST (IER,6HVMULFF)
C       CALL USWFM (14HGAIN MATRIX F:,14,F,10,NI,NS,4)    !**
        CALL VMULFF (B,F,NS,NI,NS,10,10,AHAT,10,IER)
        CALL UERTST (IER,6HVMULFF)
C       CALL USWFM (4HB*F:,4,AHAT,10,NS,NS,4)             !**
        DO 240 II=1,NS
        DO 240 IJ=1,NS
        AHAT(II,IJ)=AHAT(II,IJ)+A(II,IJ)
 240    CONTINUE
C       CALL USWFM (12HMATRIX AHAT:,12,AHAT,10,NS,NS,4)    !**
C       WRITE (5,2)
C   2   FORMAT (1X,'EXITING SUBROUTINE GAIN ==================')
        RETURN
        END
```

```
C*********************************************************************
C*********************************************************************
        SUBROUTINE MODE8
C-Function: Facilitates storage and handelling of CURRENT data.
C-IMSL routines called: UERSET.
C-Spectral Assignment routines: -.
C-Logical devices; Input Unit:   5   Output Unit:   5
C               Storage Unit(s): IU=20,IUT=20+ns+1,IBAK=IUT+I for I=1,9.
C-Random Access Files: SYSTEM.DAT,CURRNT.DAT,FORxx.DAT where xx=IBAK.
        REAL AUX1(10,10),AUX2(10,10),AUX3(10,10)
      REAL XX(10,10),VA(20),E(20),X(20),WJ(10)
      REAL W(10,10),V(10,10),VINV(10,10),F(10,10),AHAT(10,10)
        REAL A(10,10),B(10,10),C(10,10)
        COMMON/SYS/A,B,C,ZERO,IDGT,NS,NI,NO
        COMMON/AUG/F,AHAT/AUX/AUX1,AUX2,AUX3
        COMMON/VEC/VA,E,X,WJ,W,XX,V,VINV
        CALL UERSET(3,LEVOLD)
        IU=20
        READ (IU'1) NS,NI,NO,IDGT,ZERO
        WRITE (5,11)
   11   FORMAT (1X,23(1H*),22H MODE 8:DATA TRANSFER ,25(1H*),//,
     11X,54HENTER # OF BACKUP FILE YOU WISH TO ADDRESS,I=1,...,9 :)
        READ (5,*) I
        WRITE (5,12)
   12 FORMAT (1X,48HSET TRANSFER OPTIONS:--1 FOR [CURRNT]==>[BAKUPI],/
     1,22X,27H--2 FOR [CURRNT]<==[BAKUPI],/
     2,22X,28H--3 FOR [CURRNT]<==>[BAKUPI])
        READ (5,*) IOP
        IUT=20+NS+1
      OPEN (FILE='CURRNT.DAT',ACCESS='RANDOM',RECORD SIZE=102
     1,UNIT=IUT,MODE='BINARY',DEVICE='DSK',DISPOSE='SAVE')
        IBAK=IUT+I
      OPEN (ACCESS='RANDOM',RECORD SIZE=102
     1,UNIT=IBAK,MODE='BINARY',DEVICE='DSK',DISPOSE='SAVE')
        IF (IOP.EQ.2) GO TO 20
        READ (IUT'1) ((V(II,IJ),IJ=1,NS),II=1,NS)
        READ (IUT'2) ((XX(II,IJ),IJ=1,NS),II=1,NI)
        READ (IUT'4) ((F(II,IJ),IJ=1,NS),II=1,NI)
        READ (IUT'5) ((AHAT(II,IJ),IJ=1,NS),II=1,NS)
C       CALL USWFM (10HMATRIX V :,10,V,10,NS,NS,4)         !**
C       CALL USWFM (10HMATRIX XX:,10,XX,10,NI,NS,4)        !**
C       CALL USWFM (14HGAIN MATRIX F:,14,F,10,NI,NS,4)     !**
C       CALL USWFM (12HMATRIX AHAT:,12,AHAT,10,NS,NS,4)    !**
        IF (IOP.EQ.3) GO TO 30
        WRITE (IBAK'1) ((V(II,IJ),IJ=1,NS),II=1,NS)
        WRITE (IBAK'2) ((XX(II,IJ),IJ=1,NS),II=1,NI)
        WRITE (IBAK'4) ((F(II,IJ),IJ=1,NS),II=1,NI)
        WRITE (IBAK'5) ((AHAT(II,IJ),IJ=1,NS),II=1,NS)
        GO TO 999
   30   DO 34 II=1,NS
        DO 34 IJ=1,NS
        AUX1(II,IJ)=V(II,IJ)
        AUX2(II,IJ)=AHAT(II,IJ)
   34   CONTINUE
        READ (IBAK'1) ((V(II,IJ),IJ=1,NS),II=1,NS)
        READ (IBAK'5) ((AHAT(II,IJ),IJ=1,NS),II=1,NS)
        WRITE (IUT'1) ((V(II,IJ),IJ=1,NS),II=1,NS)
        WRITE (IUT'5) ((AHAT(II,IJ),IJ=1,NS),II=1,NS)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      WRITE (IBAK'1) ((AUX1(II,IJ),IJ=1,NS),II=1,NS)
      WRITE (IBAK'5) ((AUX2(II,IJ),IJ=1,NS),II=1,NS)
      DO 35 II=1,NI
      DO 35 IJ=1,NS
      AUX2(II,IJ)=XX(II,IJ)
      AUX3(II,IJ)=F(II,IJ)
   35 CONTINUE
      READ (IBAK'2) ((XX(II,IJ),IJ=1,NS),II=1,NI)
      READ (IBAK'4) ((F(II,IJ),IJ=1,NS),II=1,NI)
      WRITE (IUT'2) ((XX(II,IJ),IJ=1,NS),II=1,NI)
      WRITE (IUT'4) ((F(II,IJ),IJ=1,NS),II=1,NI)
      WRITE (IBAK'2) ((AUX2(II,IJ),IJ=1,NS),II=1,NI)
      WRITE (IBAK'4) ((AUX3(II,IJ),IJ=1,NS),II=1,NI)
      GO TO  999
   20 READ (IBAK'1) ((V(II,IJ),IJ=1,NS),II=1,NS)
      READ (IBAK'2) ((XX(II,IJ),IJ=1,NS),II=1,NI)
      READ (IBAK'4) ((F(II,IJ),IJ=1,NS),II=1,NI)
      READ (IBAK'5) ((AHAT(II,IJ),IJ=1,NS),II=1,NS)
      WRITE (IUT'1) ((V(II,IJ),IJ=1,NS),II=1,NS)
      WRITE (IUT'2) ((XX(II,IJ),IJ=1,NS),II=1,NI)
      WRITE (IUT'4) ((F(II,IJ),IJ=1,NS),II=1,NI)
      WRITE (IUT'5) ((AHAT(II,IJ),IJ=1,NS),II=1,NS)
  999 GO TO (1,2,3),IOP
    1 WRITE (5,13) I
   13 FORMAT (10X,17H[CURRNT]==>[BAKUP,I1,1H])
      GO TO 900
    2 WRITE (5,14) I
   14 FORMAT (10X,17H[CURRNT]<==[BAKUP,I1,1H])
      GO TO 900
    3 WRITE (5,15) I
   15 FORMAT (10X,18H[CURRNT]<==>[BAKUP,I1,1H])
  900 RETURN
      END
```

```
C*****************************************************************
C*****************************************************************
      SUBROUTINE DSPLAY(NS,ZERO)
C-Function: Displays normalized Eigenvectos.
C-IMSL routines called: USWFM.
C-Spectral Assignment routines: NORM.
C-Logical devices; Input Unit:   -   Output Unit:   5
C              Storage Unit(s): IU=20+J for J=1,NS.
C-Random Access Files: FORxx.DAT where xx=20+J for J=1,NS.
      REAL MAT(10,10),LRE(10),LIM(10),VA(20),E(20),X(20),WJ(10)
      PEAL W(10,10),XX(10,10),V(10,10),VINV(10,10)
      COMMON/VEC/VA,E,X,WJ,W,XX,V,VINV/EIG/LRE,LIM
      J=1
   10 IU=J+20
      READ (IU'1) LRE(J),LIM(J)
      IF (ABS(LIM(J)).GT.ZERO) GO TO 100
      DO 20 I=1,NS
      VA(I)=V(I,J)
   20 CONTINUE
      CALL NORM(VA,NS,XVA)
      DO 30 I=1,NS
      MAT(I,J)=VA(I)/XVA
   30 CONTINUE
      GO TO 200
  100 NS2=2*NS
      JC=J+1
      DO 120 I=1,NS2
      IF (I.GT.NS) GO TO 110
      VA(I)=V(I,J)
      GO TO 120
  110 INS=I-NS
      VA(I)=V(INS,JC)
  120 CONTINUE
      CALL NORM(VA,NS2,XVA)
      DO 140 I=1,NS2
      IF (I.GT.NS) GO TO 130
      MAT(I,J)=VA(I)/XVA
      GO TO 140
  130 INS=I-NS
      MAT(INS,JC)=VA(I)/XVA
  140 CONTINUE
      J=J+1
  200 IF(J.GE.NS) GO TO 300
      J=J+1
      GO TO 10
  300 CALL USWFM(20HNORMALIZED VECTORS :,20,MAT,10,NS,NS,4)
      RETURN
      END
```

```
C*****************************************************************
C*****************************************************************
      SUBROUTINE MODE4
C-Function: Simulates and plots time responses.
C-IMSL routines called: UERSET,UERTST,UGETIO,DVERK,VMULFF,USPLO.
C-Spectral Assignment routines: UEVAL,FCN.
C-Logical devices; Input Unit:   5   Output Unit:   5
C              Storage Unit(s): IU=20,IUT=20+NS+1.
C-Random Access Files: SYSTEM.DAT,CURRNT.DAT .
      INTEGER INOPT(10)
      REAL AMP(10),SLOPE(10),U(10),ATIL(10,10),CCNST(10)
      REAL X(10),CX(24),W(10,10),Y(10),XPRIME(10)
C PLOT ARRAYES
      REAL T(201),UMAT(201,10),XMAT(201,10),VEC(201),RANGE(4)
      REAL IMAG4(5151)
C SYSTEM ARRAYES
      REAL A(10,10),B(10,10),C(10,10),F(10,10),AHAT(10,10)
      COMMON/SYS/A,B,C,ZERO,IDGT,NS,NI,NO
      COMMON/AUG/F,AHAT
      COMMON/DIF/ATIL,CONST
      EXTERNAL FCN
      CALL UERSET(3,LEVOLD)
C****************** READ SYSTEM DATA *************************
      IRS=102
      IU=20
      READ (IU'1) NS,NI,NO,IDGT,ZERO
      READ (IU'2) ((A(I,J),J=1,NS),I=1,NS)
      READ (IU'3) ((B(I,J),J=1,NI),I=1,NS)
      READ (IU'4) ((C(I,J),J=1,NS),I=1,NO)
      IUT=IU+NS+1
     OPEN(FILE='CURRNT.DAT',ACCESS='RANDOM',RECORD SIZE=IRS
    1,UNIT=IUT,MODE='BINARY',DEVICE='DSK',DISPOSE='SAVE')
      READ (IUT'4) ((F(I,J),J=1,NS),I=1,NI)
      READ (IUT'5) ((AHAT(I,J),J=1,NS),I=1,NS)
C
 180  WRITE (5,11)
  11  FORMAT (1X,23(1H*),24H MODE 4:TIME SIMULATION ,23(1H*),//,1X,10(1H
     1*),27H CHOOSE SIMULATION OPTIONS:,/,1X,1H-,62HENTER: 1 TO SIMULATE
     2 [A], 2 TO SIMULATE [AHAT],(3 FOR [ATIL]):)
 177  CONTINUE
      DO 178 II=1,201
      T(II)=FLOAT(0)
      VEC(II)=FLOAT(0)
 178  CONTINUE
      READ (5,*) ISYS
      GO TO (1,2,3),ISYS
   1  DO 10 I=1,NS
      DO 10 J=1,NS
      ATIL(I,J)=A(I,J)
  10  CONTINUE
      GO TO 30
   2  DO 20 I=1,NS
      DO 20 J=1,NS
      ATIL (I,J)=AHAT(I,J)
  20  CONTINUE
      GO TO 30
```

```
    3     WRITE (5,12)
   12     FORMAT (1X,36HENTER SYSTEM MATRIX TO BE SIMULATED:)
          READ (5,*) ((ATIL(I,J),J=1,NS),I=1,NS)
   30     CONTINUE
          WRITE (5,13)
   13  FORMAT (1X,58HENTER 0 TO SIMULATE OUTPUTS,1 TO SIMULATE STATE VARI
      1ABLES:)
          READ (5,*) IOUT
          WRITE (5,14)
   14  FORMAT (1X,47HENTER SIMULATION TIME,(REAL NUMBER IN SECONDS):)
          READ (5,*) DT
          WRITE (5,15)
   15  FORMAT (1X,50HENTER NUMBER OF POINTS TO BE CALCULATED,(200 MAX):)
          READ (5,*) NP
          WRITE (5,16)
   16  FORMAT (1X,31HSPECIFY THE INITIAL CONDITIONS:)
          DO 40 I=1,NS
          WRITE (5,17) I
   17  FORMAT (1X,1HX,I2,4H(0):)
          READ (5,*) X(I)
   40     CONTINUE
          WRITE (5,32)
   32  FORMAT (1X,56HCHOOSE INPUT OPTIONS:1 FOR NO INPUT, 2 FOR A STEP IN
      1PUT,/,1X,21(1H ),40H3 FOR A RAMP,AND 4 FOR A TRUNCATED RAMP:)
          DO 50 I=1,NI
          WRITE (5,18) I
   18  FORMAT (1X,18HINPUT OPTION FOR U,I2,1H:)
          READ (5,*) INOPT(I)
          IF (INOPT(I).NE.2) GO TO 51
          WRITE (5,19) I
   19  FORMAT (1X,37HSPECIFY AMPLITUDE OF THE STEP INPUT U,I2,1H:)
          READ (5,*) AMP(I)
          GO TO 50
   51     IF (INOPT(I).NE.3) GO TO 52
          WRITE (5,21) I
   21  FORMAT (1X,33HSPECIFY SLOPE OF THE RAMP INPUT U,I2,1H:)
          READ (5,*) SLOPE(I)
          GO TO 50
   52     IF (INOPT(I).NE.4) GO TO 50
          WRITE (5,22) I
   22  FORMAT (1X,33HSPECIFY AMPLITUDE AND SLOPE FOR U,I2,1H:)
          READ (5,*) AMP(I),SLOPE(I)
   50     CONTINUE
C*************** DIFFERENTIAL EQUATION SOLUTION ***************
          IND=1
          TOL=ZERO*100.000000
          TINT=DT/NP
          NP1=NP+1
          DO 100 K=1,NP1
          KM1=K-1
          TEND=FLOAT(KM1)*TINT
          CALL UEVAL (INOPT,AMP,SLOPE,U,NI,TEND)
          CALL VMULFF (B,U,NS,NI,1,10,10,CONST,10,IER)
C         CALL UERTST (IER,6HVMULFF)                        !**
          CALL DVERK (NS,FCN,T,X,TEND,TOL,IND,CX,10,W,IER)
          IF (IND.LT.0.OR.IER.GT.0) GO TO 190
```

```
   53     T(K)=TEND
          DO 60 J=1,NI
          UMAT(K,J)=U(J)
   60     CONTINUE
          IF (IOUT.EQ.0) GO TO 80
          DO 70 J=1,NS
          XMAT(K,J)=X(J)
   70     CONTINUE
          N=NS
          GO TO 100
   80     CALL VMULFF (C,X,NO,NS,1,10,10,Y,10,IER)
C         CALL UERTST (IER,6HVMULFF)                        !**
          DO 90 J=1,NO
          XMAT(K,J)=Y(J)
   90     CONTINUE
          N=NO
  100     CONTINUE .
C******************** PLOT *****************************************
          WRITE (5,23)
   23  FORMAT (1X,49HENTER 0 FOR 80 DISPLAY COLUMNS,1 FOR 129 COLUMNS:)
          READ (5,*) IOPT
  115     WRITE (5,24)
   24  FORMAT (1X,48HENTER 0 FOR INDIVIDUAL AND 1 FOR MULTIPLE PLOTS:)
          READ (5,*) IPLOT
          WRITE (5,25)
   25  FORMAT (1X,51HDO YOU WISH TO SET THE MIN-MAX RANGES FOR THE AXES?)
          READ (5,*) IRANGE
          IF (IRANGE.GT.0) GO TO 120
          DO 110 I=1,4
          RANGE(I)=0.0
  110     CONTINUE
          GO TO 124
  120     WRITE (5,26)
   26  FORMAT (1X,41HENTER MIN X,MAX X,MIN Y,AND MAX Y VALUES:)
          READ (5,*) (RANGE(I),I=1,4)
C*************** PLOT INPUTS ***************************************
  124     DO 125 J=1,NI
          IF (INOPT(J).NE.1) GO TO 130
  125     CONTINUE
          GO TO 135
  130     CONTINUE
          WRITE (5,33)
   33  FORMAT (1X,50HPOSITION PAPER AT TOP OF FORM AND TYPE ANY INTEGER,/
      1,1X,41HYOU MAY ADD A SHORT NOTE (20 CHARACTERS.))
          READ (5,34) III
   34     FORMAT (I1,20X)
       CALL USPLO (T,UMAT,201,NP,NI,1,13HSYSTEM INPUTS,13,4HTIME,4
      1,5HINPUT,5,RANGE,10H1234567890,IOPT,IER)
          CALL UERTST (IER,6HUSPLO )
  135     IF (IPLOT.LE.0) GO TO 140
C*************** PLOT STATE VARIABLES OR OUTPUTS *****************
          WRITE (5,33)
          READ (5,34) III
          CALL USPLO (T,XMAT,201,NP,N,1,15HTIME SIMULATION,15,4HTIME,4
      1,8HRESPONSE,8,RANGE,10H1234567890,IOPT,IER)
          CALL UERTST (IER,6HUSPLO )
```

```
            GO TO 170
     140    DO 160 J=1,N
            DO 150 I=1,NP
            VEC(I)=XMAT(I,J)
     150    CONTINUE
            WRITE (5,33)
            READ (5,34) III
            CALL USPLO (T,VEC,201,NP,1,1,15HTIME SIMULATION,15,4HTIME,4
         1,8HRESPONSE,8,RANGE,1HX,IOPT,IER)
            CALL UERTST (IER,6HUSPLO )
     160    CONTINUE
     170    WRITE (5,27)
      27  FORMAT (1X,28HWISH TO REPEAT THE PLOTTING?)
            READ (5,*) K1
            IF (K1.GT.0) GO TO 115
            WRITE (5,28)
      28  FORMAT (1X,28HWISH TO EXIT FROM THIS MODE?)
            READ (5,*) K2
            IF (K2.LE.0) GO TO 180
            WRITE (5,29)
      29  FORMAT (1X,27(1H*),18H  EXITING MODE 4   ,25(1H*))
            GO TO 200
     190    WRITE (5,31) IND,IER,K
      31  FORMAT (1X,4HIND=,I2,4HIER=,I3,51HCHECK INSTRUCTIONS FOR DIAGNOSTI
         1C MESSAGES ON DVERK,/,1X,28HPROBLEM ON ITERATION NUMBER ,I3)
            GO TO 53
     200    RETURN
            END
```

```
C******************************************************************
C******************************************************************
      SUBROUTINE UEVAL(INOPT,AMP,SLOPE,U,NI,TEND)
C-Function: Evaluates the input forcing functions.
C-IMSL routines called: -
C-Spectral Assignment routines: -
C-Logical devices; Input Unit:   -   Output Unit:   -
C             Storage Unit(s): -
C-Random Access Files: -
      INTEGER INOPT(NI)
      REAL AMP(NI),SLOPE(NI),U(NI),TEND
      DO 10 I=1,NI
      GO TO (1,2,3,4),INOPT(I)
    1 U(I)=0.000000
      GO TO 10
    2 U(I)=AMP(I)
      GO TO 10
    3 U(I)=SLOPE(I)*TEND
      GO TO 10
    4 IF (TEND.LE.(AMP(I)/SLOPE(I))) GO TO 3
      GO TO 2
   10 CONTINUE
      RETURN
      END
```

```
C*************************************************************
C*************************************************************
      SUBROUTINE FCN(NS,T,X,XPRIME)
C-Function: Evaluates x' fuctions for use by IMSL routine DVERK.
C-IMSL routines called: -
C-Spectral Assignment routines: -
C-Logical devices; Input Unit:   -   Output Unit:   -
C              Storage Unit(s): -
C-Random Access Files: -
      REAL X(NS),XPRIME(NS),ATIL(10,10),CONST(10)
      COMMON/DIF/ATIL,CONST
      DO 10 I=1,NS
     XPRIME(I)=ATIL(I,1)*X(1)+ATIL(I,2)*X(2)+ATIL(I,3)*X(3)+ATIL(I,4)*X
    1(4)+ATIL(I,5)*X(5)+ATIL(I,6)*X(6)+ATIL(I,7)*X(7)+ATIL(I,8)*X(8)+AT
    2IL(I,9)*X(9)+ATIL(I,10)*X(10)+CONST(I)
   10 CONTINUE
      RETURN
      END
```

```
C**************************************************************
C**************************************************************
            SUBROUTINE MODE5
C-Function: Main routine for Component Modification
C-IMSL routines called: UERSET,USWFM.
C-Spectral Assignment routines: CGRAD,CCOST,SEARCH,DSPLAY.
C-Logical devices; Input Unit:   5    Output Unit:   5
C-             Storage Unit(s): IU=20,IU=20+J for J=1,NS,IUT=20+NS+1
C-Random Access Files: SYSTEM.DAT,FORxx.DAT where xx=20+J ,CURRNT.DAT
            REAL AL(10,10),G(10,10)
         REAL XX(10,10),VA(20),E(20),X(20),LRE(10),LIM(10),WJ(10)
         REAL W(10,10),V(10,10),VINV(10,10),F(10,10),AHAT(10,10)
            REAL A(10,10),B(10,10),C(10,10)
            COMMON/SYS/A,B,C,ZERO,IDGT,NS,NI,NO
            COMMON/AUG/F,AHAT/EIG/LRE,LIM/PAR/AL/GR/G
            COMMON/VEC/VA,E,X,WJ,W,XX,V,VINV
            COMMON/COMP/IROW,ICOL,F1,F2
            EXTERNAL CCOST,CGRAD
            CALL UERSET(3,LEVOLD)
            IU=20
            READ (IU'1) NS,NI,NO,IDGT,ZERO
            READ (IU'2) ((A(II,IJ),IJ=1,NS),II=1,NS)
            READ (IU'3) ((B(II,IJ),IJ=1,NI),II=1,NS)
            DO 10 J=1,NS
            IU=20+J
         OPEN (ACCESS='RANDOM',RECORD SIZE=202
        1,UNIT=IU,MODE='BINARY',DEVICE='DSK',DISPOSE='SAVE')
            READ (IU'1) LRE(J),LIM(J)
   10       CONTINUE
            IUT=20+NS+1
         OPEN (FILE='CURRNT.DAT',ACCESS='RANDOM',RECORD SIZE=102
        1,UNIT=IUT,MODE='BINARY',DEVICE='DSK',DISPOSE='SAVE')
            READ (IUT'1) ((V(II,IJ),IJ=1,NS),II=1,NS)
            READ (IUT'2) ((XX(II,IJ),IJ=1,NS),II=1,NI)
            READ (IUT'4) ((F(II,IJ),IJ=1,NS),II=1,NI)
            READ (IUT'5) ((AHAT(II,IJ),IJ=1,NS),II=1,NS)
            CALL USWFM (10HMATRIX V :,10,V,10,NS,NS,4)         !**
C           CALL USWFM (10HMATRIX XX:,10,XX,10,NI,NS,4)        !**
C           CALL USWFM (14HGAIN MATRIX F:,14,F,10,NI,NS,4)     !**
C           CALL USWFM (12HMATRIX AHAT:,12,AHAT,10,NS,NS,4)    !**
            DO 20 II=1,NS
            DO 20 IJ=1,NS
            AL(II,IJ)=V(II,IJ)
   20       CONTINUE
            WRITE (5,1)
    1       FORMAT (1X,22(1H*),28H MODE 5:COMPONENT REDUCTION ,20(1H*),//,
        11X,52HENTER THE COORDINATES OF THE COMPONENT TO BE REDUCED,/,
        21X,32HROW=--,COLUMN=--(BOTH INTEGERS):)
            READ (5,*) IROW,ICOL
            WRITE (5,2)
    2       FORMAT (1X,39HSET DESIRED WEIGHTS,DEFAULT VALUES ARE:,/,
        11X,11HF1=F2=1.000,/,1X,15HWISH TO CHANGE?)
            READ (5,*) KK
            F1=FLOAT(1)
            F2=F1
            IF (KK.LE.0) GO TO 30
```

```
          WRITE (5,3)
    3     FORMAT (1X,17HENTER NEW VALUES:)
          READ (5,*) F1,F2
   30     CALL CCOST(CJ)
          WRITE (5,4) CJ                                    !**
    4     FORMAT (1X,5HCOST=,E15.6)                   !**
          CALL CGRAD
          CALL SEARCH(CJ,CCOST,CGRAD,5)
          WRITE (IUT'1) ((V(II,IJ),IJ=1,NS),II=1,NS)
          WRITE (IUT'2) ((XX(II,IJ),IJ=1,NS),II=1,NI)
          WRITE (IUT'4) ((F(II,IJ),IJ=1,NS),II=1,NI)
          WRITE (IUT'5) ((AHAT(II,IJ),IJ=1,NS),II=1,NS)
          CALL USWFM (10HMATRIX V :,10,V,10,NS,NS,4)        !**
          WRITE (5,902)
  902   FORMAT (1X,44HWISH TO DISPLAY THE NORMALIZED EIGENVECTORS?)
          READ (5,*) KS
          IF (KS.LE.0) GO TO 903
          CALL DSPLAY (NS,ZERO)
  903     CONTINUE
C         CALL USWFM (10HMATRIX XX:,10,XX,10,NI,NS,4)       !**
          CALL USWFM (14HGAIN MATRIX F:,14,F,10,NI,NS,4)    !**
C         CALL USWFM (12HMATRIX AHAT:,12,AHAT,10,NS,NS,4)   !**
          RETURN
          END
```

```
C*****************************************************************
C*****************************************************************
          SUBROUTINE CCOST(CJ)
C-Function: Calculates the COST function for component modification.
C-IMSL routines called: -
C-Spectral Assignment routines: -
C-Logical devices; Input Unit:    -    Output Unit:   5
C-             Storage Unit(s): -
C-Random Access Files: -
         REAL XX(10,10),VA(20),E(20),X(20),LRE(10),LIM(10),WJ(10)
         REAL W(10,10),V(10,10),VINV(10,10),AL(10,10)
          REAL A(10,10),B(10,10),C(10,10)
          COMMON/SYS/A,B,C,ZERO,IDGT,NS,NI,NO
          COMMON/VEC/VA,E,X,WJ,W,XX,V,VINV
          COMMON/COMP/IROW,ICOL,F1,F2/EIG/LRE,LIM/PAR/AL
          ICOL1=ICOL+1
          CJ1=F1*V(IROW,ICOL)**2
          IF (ABS(LIM(ICOL)).GT.ABS(ZERO)) CJ1=CJ1+F1*V(IROW,ICOL1)**2
          CJ2=FLOAT(0)
          N=1
   10     N1=N+1
          DO 100 M=1,NS
          IF (N.EQ.ICOL.AND.M.EQ.IROW) GO TO 100
          CJ2=CJ2+((V(M,N)-AL(M,N))**2)*F2
          IF(ABS(LIM(N)).GT.ABS(ZERO)) CJ2=CJ2+(V(M,N1)-AL(M,N1))**2*F2
  100     CONTINUE
          N=N+1
          IF (ABS(LIM(N)).GT.ABS(ZERO)) N=N+1
          IF (N.LE.NS) GO TO 10
          CJ=CJ1+CJ2
          WRITE (5,1) CJ1,CJ2
    1     FORMAT (20X,4H J1=,E15.6,5X,4H J2=,E15.6)
          RETURN
          END
```

```
C****************************************************************
C****************************************************************
      SUBROUTINE CGRAD
C-Function: Calculates the GRADIENT for component modification.
C-IMSL routines called: USWFM
C-Spectral Assignment routines: PVP,DBNORM.
C-Logical devices; Input Unit:  -    Output Unit:   5
C-         Storage Unit(s): -
C-Random Access Files: -
      REAL G(10,10),PJ1(10,10),PJ2(10,10)
    REAL XX(10,10),VA(20),E(20),X(20),LRE(10),LIM(10),WJ(10)
    REAL W(10,10),V(10,10),VINV(10,10),AL(10,10)
      REAL A(10,10),B(10,10),C(10,10)
      REAL AUX1(10,10),AUX2(10,10),AUX3(10,10)
      COMMON/SYS/A,B,C,ZERO,IDGT,NS,NI,NO
      COMMON/VEC/VA,E,X,WJ,W,XX,V,VINV/AUX/AUX1,AUX2,AUX3
      COMMON/COMP/IROW,ICOL,F1,F2/EIG/LRE,LIM/PAR/AL/GR/G/PJ/PJ1,PJ2
      ICOL1=ICOL+1
      J=1
   10 J1=J+1
      DO 105 I=1,NI
      KI=I
      KJ=J
      CALL PVP(KI,KJ)
      IF (ICOL.NE.J) GO TO 14
      PJ1(I,J)=2*F1*V(IROW,ICOL)*AUX1(IROW,ICOL)
      IF (ABS(LIM(ICOL)).LE.ABS(ZERO)) GO TO 15
C     CALL USWFM(8HPV/PXIJ:,8,AUX1,10,NS,NS,4)   !**
C     CALL USWFM(10HPV/PXIJ+1:,10,AUX2,10,NSINS,4)   !**
      PJ1(I,J)=PJ1(I,J)+2*F1*V(IROW,ICOL1)*AUX1(IRNW,ICOL1)
    PJ1(I,J1)=2*F1*(V(IROW,ICOL)*AUX2(IROW,ICOL1)+V(IROW,ICOL1)*
   1AUX2(IROW,ICOL1))
      GO TO 15
   14 PJ1(I,J)=FLOAT(0)
      IF (ABS(LIM(ICOL)).GT.ABS(ZERO)) PJ1(I,J1)=FLOAT(0)
   15 PJ2(I,J)=FLOAT(0)
      IF (ABS(LIM(J)).GT.ABS(ZERO)) PJ2(I,J1)=FLOAT(0)
      DO 100 M=1,NS
      IF (J.EQ.ICOL.AND.M.EQ.IROW) GO TO 100
      PJ2(I,J)=PJ2(I,J)+(AL(M,J)-V(M,J))*AUX1(M,J)
      IF (ABS(LIM(J)).LE.ABS(ZERO)) GO TO 100
      PJ2(I,J)=PJ2(I,J)+(AL(M,J1)-V(M,J1))*AUX1(M,J1)
      PJ2(I,J1)=PJ2(I,J1)+(AL(M,J)-V(M,J))*AUX2(M,J)+
   1(AL(M,J1)-V(M,J1))*AUX2(M,J1)
  100 CONTINUE
      PJ2(I,J)=2*F2*PJ2(I,J)
      IF (ABS(LIM(J)).GT.ABS(ZERO)) PJ2(I,J1)=2*F2*PJ2(I,J1)
  105 CONTINUE
      IF (ABS(LIM(J)).GT.ABS(ZERO)) J=J+1
      J=J+1
      IF (J.LE.NS) GO TO 10
      DO 110 II=1,NI
      DO 110 IJ=1,NS
      G(II,IJ)=PJ1(II,IJ)+PJ2(II,IJ)
  110 CONTINUE
C     CALL USWFM(11HMATRIX [G]:,11,G,10,NI,NS,4)   !**
      CALL DBNORM(NI,NS)
      CALL USWFM(16HGRADIENT MATRIX:,16,G,10,NI,NS,4)   !**
      RETURN
      END
```

```
C***************************************************************
C***************************************************************
      SUBROUTINE PVP(KI,KJ)
C-Function: Returns p[V]/p[X]ij .
C-IMSL routines called: (USWFM).
C-Spectral Assignment routines: -
C-Logical devices; Input Unit:   -    Output Unit:   (5)
C-      Storage Unit(s): IU=20+KJ ,KJ specified by CALL statement.
C-Random Access Files: FORxx.DAT where xx=20+KJ.
      REAL AUX1(10,10),AUX2(10,10),AUX3(10,10)
C NULL SPACE ARRAYS
.     REAL ML(10,10),NL(10,10),ALPHA(20,20),BETA(20,20)
      REAL NLC(10,20),PLC(10,20),MLC(10,20)
      REAL STAR(20,20),QL(10,20),RL(10,20)
C MODE 3 ARRAYS
     REAL XX(10,10),VA(20),E(20),X(20),LRE(10),LIM(10),WJ(10)
     REAL W(10,10),V(10,10),VINV(10,10),F(10,10),AHAT(10,10)
      REAL A(10,10),B(10,10),C(10,10)
      COMMON/SYS/A,B,C,ZERO,IDGT,NS,NI,NO
      COMMON/VEC/VA,E,X,WJ,W,XX,V,VINV/EIG/LRE,LIM
      COMMON/NSPA/ML,NL,NLC,PLC,MLC,STAR,QL,RL
      COMMON/AUX/AUX1,AUX2,AUX3
      IU=20+KJ
      IF (ABS(LIM(KJ)).GT.ABS(ZERO)) GO TO20
      READ (IU'3) ((NL(II,IJ),IJ=1,NI),II=1,NS)
      DO 10 I=1,NS
      DO 10 J=1,NS
      AUX1(I,J)=FLOAT(0)
      IF (J.EQ.KJ) AUX1(I,J)=NL(I,KI)
      AUX2(I,J)=FLOAT(0)
   10 CONTINUE
      GO TO 30
   20 NI2=NI*2
      READ (IU'6) ((QL(II,IJ),IJ=1,NI2),II=1,NS)
      READ (IU'7) ((RL(II,IJ),IJ=1,NI2),II=1,NS)
C     CALL USWFM (10HMATRIX QL:,10,QL,10,NS,NI2,4)      !**
C     CALL USWFM (10HMATRIX RL:,10,RL,10,NS,NI2,4)      !**
      KIN=KI+NI
      KJ1=KJ+1
      DO 30 I=1,NS
      DO 30 J=1,NS
      AUX1(I,J)=FLOAT(0)
      AUX2(I,J)=FLOAT(0)
      IF(J.EC.KJ) AUX1(I,J)=QL(I,KI)
      IF(J.EQ.KJ1) AUX1(I,J)=RL(I,KI)
      IF(J.EQ.KJ) AUX2(I,J)=QL(I,KIN)
      IF(J.EQ.KJ1) AUX2(I,J)=RL(I,KIN)
   30 CONTINUE
C     WRITE (5,1) KI,KJ                 !**
C  1  FORMAT (1X,'I=',I2,'J=',I2)       !**
C     CALL USWFM(8HPV/PXIJ:,8,AUX1,10,NS,NS,4)   !**
C     CALL USWFM(10HPV/PXIJ+1:,10,AUX2,10,NS,NS,4)   !**
      RETURN
      END
```

```fortran
C*****************************************************************
C*****************************************************************
      SUBROUTINE MODE6
C-Function: Main routine for Gain Modification.
C-IMSL routines called: UERSET,USWFM.
C-Spectral Assignment routines: GCOST,GGRAD,SEARCH,DSPLAY.
C-Logical devices; Input Unit:   5    Output Unit:   5
C-            Storage Unit(s): IU=20,IUT=20+NS+1
C-Random Access Files: SYSTEM.DAT,CURRNT.DAT .
C GRADIENT ARRAYAS
      REAL AL(10,10),G(10,10),AUX1(10,10),AUX2(10,10),AUX3(10,10)
C NULL SPACE ARRAYS
      REAL ML(10,10),NL(10,10),ALPHA(20,20),BETA(20,20)
      REAL NLC(10,20),PLC(10,20),MLC(10,20)
      REAL STAR(20,20),QL(10,20),RL(10,20)
C MODE 3 ARRAYS
     REAL XX(10,10),VA(20),E(20),X(20),LRE(10),LIM(10),WJ(10)
     REAL W(10,10),V(10,10),VINV(10,10),F(10,10),AHAT(10,10)
      REAL A(10,10),B(10,10),C(10,10)
      COMMON/SYS/A,B,C,ZERO,IDGT,NS,NI,NO
      COMMON/AUG/F,AHAT/EIG/LRE,LIM/PAR/AL/GR/G
      COMMON/VEC/VA,E,X,WJ,W,XX,V,VINV
      COMMON/NSPA/ML,NL,NLC,PLC,MLC,STAR,QL,RL
      COMMON/AUX/AUX1,AUX2,AUX3
      EXTERNAL GCOST,GGRAD
      CALL UERSET(3,LEVOLD)
      IU=20
      READ (IU'1) NS,NI,NO,IDGT,ZERO
      READ (IU'2) ((A(II,IJ),IJ=1,NS),II=1,NS)
      READ (IU'3) ((B(II,IJ),IJ=1,NI),II=1,NS)
      IUT=20+NS+1
     OPEN (FILE='CURRNT.DAT',ACCESS='RANDOM',RECORD SIZE=102
    1,UNIT=IUT,MODE='BINARY',DEVICE='DSK',DISPOSE='SAVE')
      READ (IUT'1) ((V(II,IJ),IJ=1,NS),II=1,NS)
      READ (IUT'2) ((XX(II,IJ),IJ=1,NS),II=1,NI)
      READ (IUT'4) ((F(II,IJ),IJ=1,NS),II=1,NI)
      READ (IUT'5) ((AHAT(II,IJ),IJ=1,NS),II=1,NS)
      CALL USWFM (10HMATRIX V :,10,V,10,NS,NS,4)        !**
C     CALL USWFM (10HMATRIX XX:,10,XX,10,NI,NS,4)       !**
      CALL USWFM (14HGAIN MATRIX F:,14,F,10,NI,NS,4)    !**
C     CALL USWFM (12HMATRIX AHAT:,12,AHAT,10,NS,NS,4)   !**
      DO 10 II=1,NI
      DO 10 IJ=1,NS
      AL(II,IJ)=FLOAT(1)
 10   CONTINUE
      WRITE (5,1)
  1   FORMAT (1X,22(1H*),23H MODE 6:GAIN REDUCTION ,25(1H*),//,
    11X,22HSET ALPHA PARAMETERS :,/,1X,20HDEFAULT VALUES ARE :)
      CALL USWFM (17HGAIN PARAMETERS :,17,AL,10,NI,NS,4)   !**
      WRITE (5,2)
  2   FORMAT (1X,15HWISH TO CHANGE:)
      READ (5,*) KK
      IF (KK.LE.0) GO TO 20
      WRITE (5,4)
  4   FORMAT (1X,17HENTER NEW VALUES:)
      READ (5,*) ((AL(II,IJ),IJ=1,NS),II=1,NI)
```

```
   20    CALL GCOST(CJ)
         WRITE (5,3) CJ                                          !**
    3    FORMAT (1X,5HCOST=,E15.6)
         CALL GGRAD
         CALL SEARCH(CJ,GCOST,GGRAD,6)
         WRITE (IUT'1) ((V(II,IJ),IJ=1,NS),II=1,NS)
         WRITE (IUT'2) ((XX(II,IJ),IJ=1,NS),II=1,NI)
         WRITE (IUT'4) ((F(II,IJ),IJ=1,NS),II=1,NI)
         WRITE (IUT'5) ((AHAT(II,IJ),IJ=1,NS),II=1,NS)
         CALL USWFM (10HMATRIX V :,10,V,10,NS,NS,4)              !**
         WRITE (5,902)
  902  FORMAT (1X,44HWISH TO DISPLAY THE NORMALIZED EIGENVECTORS?)
         READ (5,*) KS
         IF (KS.LE.0) GO TO 903
         CALL DSPLAY (NS,ZERO)
  903    CONTINUE
C        CALL USWFM (10HMATRIX XX:,10,XX,10,NI,NS,4)             !**
         CALL USWFM (14HGAIN MATRIX F:,14,F,10,NI,NS,4)          !**
C        CALL USWFM (12HMATRIX AHAT:,12,AHAT,10,NS,NS,4)         !**
         RETURN
         END
```

```
C****************************************************************
C****************************************************************
            SUBROUTINE SEARCH(CJ,COST,GRAD,MODE)
C-Function: Inter-active Gradient Search Routine.
C-IMSL routines called: UERSET,UERTST,LINV2F,(USWFM),
C-Spectral Assignment routines: GAIN,COST,DESIGN,SENS,GRAD,TRAN.
C-Logical devices; Input Unit:   5     Output Unit:    5
C-              Storage Unit(s): -
C-Random Access Files: -
C GRADIENT ARRAYAS
         REAL AL(10,10),G(10,10),U(10,10),WKAREA(130)
C NULL SPACE ARRAYS
            REAL ML(10,10),NL(10,10),ALPHA(20,20),BETA(20,20)
            REAL NLC(10,20),PLC(10,20),MLC(10,20)
            REAL STAR(20,20),QL(10,20),RL(10,20)
C MODE 3 ARRAYS
         REAL XX(10,10),VA(20),E(20),X(20),LRE(10),LIM(10),WJ(10)
         REAL W(10,10),V(10,10),VINV(10,10),F(10,10),AHAT(10,10)
            REAL A(10,10),B(10,10),C(10,10)
            COMMON/SYS/A,B,C,ZERO,IDGT,NS,NI,NO
            COMMON/AUG/F,AHAT/EIG/LRE,LIM/PAR/AL/GR/G/LEG/U
            COMMON/VEC/VA,E,X,WJ,W,XX,V,VINV
            COMMON/NSPA/ML,NL,NLC,PLC,MLC,STAR,QL,RL
            CALL UERSET(3,LEVOLD)
            IFL=0
            KN=1
            N=1
            D=0.01
            DMIN=ZERO
   10       WRITE (5,1) N,D,DMIN
    1     FORMAT (1X,46HGRADIENT SEARCH ROUTINE,SET SEARCH PARAMETERS:,//
        1,1X,19HDefault values are:,/,1x,13H# of steps,N=,I3,3X,12Hstep siz
        2e,d=,E15.6,3X,5Hdmin=,E15.6,//,1x,15HWish to change?)
            READ (5,*) IUP
            IF (IUP.LE.0) GO TO 20
            WRITE (5,2)
    2       FORMAT (1X,17HEnter new values:)
            READ (5,*) N,D,DMIN
   20       IN=1
   30       DO 40 II=1,NI
            DO 40 IJ=1,NS
            XX(II,IJ)=XX(II,IJ)-D*G(II,IJ)
   40       CONTINUE
            CALL DESIGN
            CALL GAIN
            IF (MODE.NE.7) GO TO 49
            IDG=IDGT
            CALL LINV2F (V,NS,10,U,IDG,WKAREA,IER)
            CALL UERTST(IER,6HLINV2F)
C           CALL USWFM (10HMATRIX UT:,10,U,10,NS,NS,4)         !**
            CALL tran(U,NS,NS)
C           CALL USWFM (10HMATRIX U :,10,U,10,NS,NS,4)         !**
            CALL SENS
   49       CALL COST(CJNEW)
            WRITE (5,7) CJNEW                                     !**
   7        FORMAT (1X,'NEW COST=',E15.6)                         !**
```

```
      IF (CJNEW.GE.CJ) GO TO 50
      IF (IN.GE.N) GO TO 100
      CJ=CJNEW
      IN=IN+1
      KN=KN+1
      GO TO 30
   50 DO 60 II=1,NI
      DO 60 IJ=1,NS
      XX(II,IJ)=XX(II,IJ)+D*G(II,IJ)
   60 CONTINUE
      IF (KN.EQ.1) GO TO 70
      WRITE (5,3) KN,D
    3 FORMAT (1X,I3,38H Steps with present gradient and dmin=,E15.6
     1,10Hwere taken,/,1X,23HLAST STEP NOT ACCEPTED!)
      KN=1
      CALL GRAD
      GO TO 30
   70 DH=D/2
      WRITE (5,8) DH
    8 FORMAT (1X,23HLAST STEP NOT ACCEPTED!,/
     1,1X,21HSTEP SIZE REDUCED TO:,E15.6)
      IF (DH.LT.DMIN) GO TO 80
      D=DH
      GO TO 30
   80 WRITE (5,4)
    4 FORMAT (1X,36HYou are in d/2 neighborhood of Jmin!)
      IFL=1
  100 CALL DESIGN
      CALL GAIN
      WRITE (5,5) CJNEW
    5 FORMAT (1X,14HCost Function=,E15.6)
C     CALL USWFM (10HMATRIX V :,10,V,10,NS,NS,4)          !**
C     CALL USWFM (10HMATRIX XX:,10,XX,10,NI,NS,4)         !**
C     CALL USWFM (14HGAIN MATRIX F:,14,F,10,NI,NS,4)      !**
C     CALL USWFM (12HMATRIX AHAT:,12,AHAT,10,NS,NS,4)     !**
      IF (IFL.EQ.1) GO TO 90
      WRITE (5,6)
    6 FORMAT (1X,28HWish to continue the search?)
      READ (5,*)  KK
      IF (KK.LE.0) GO TO 90
      CJ=CJNEW
      GO TO 10
   90 RETURN
      END
```

```
C*******************************************************************
C*******************************************************************.
         SUBROUTINE GCOST(CJ)
C-Function: Calculates the COST function for Gain Reduction.
C-IMSL routines called: -
C-Spectral Assignment routines: -
C-Logical devices; Input Unit:   -    Output Unit:   (5)
C-         Storage Unit(s): -
C-Random Access Files: -
         REAL A(10,10),B(10,10),C(10,10)
         REAL AL(10,10),F(10,10),AHAT(10,10)
         COMMON/AUG/F,AHAT/PAR/AL
         COMMON/SYS/A,B,C,ZERO,IDGT,NS,NI,NO
         CJ=FLOAT(0)
         DO 10 I=1,NI
         DO 10 J=1,NS
         CJ=CJ+AL(I,J)*(F(I,J)**2)
  10     CONTINUE
C        WRITE (5,1) CJ                                  !**
C   1    FORMAT (1X,5HCOST=,E15.6)                           !**
         RETURN
         END
```

```
C***************************************************************
C***************************************************************
        SUBROUTINE GGRAD
C-Function: Calculates the Gradient for Gain Reduction.
C-IMSL routines called: UERTST,USWFM,LINV2F.
C-Spectral Assignment routines: DBNORM,PFX.
C-Logical devices; Input Unit:    -    Output Unit:   5
C-            Storage Unit(s): IU=20+J for J=1,NS.
C-Random Access Files: FORxx.DAT where xx=20+J .
C GRADIENT ARRAYAS
        REAL AL(10,10),G(10,10),AUX1(10,10),AUX2(10,10),AUX3(10,10)
C NULL SPACE ARRAYS
        REAL ML(10,10),NL(10,10),ALPHA(20,20),BETA(20,20)
        REAL NLC(10,20),PLC(10,20),MLC(10,20)
        REAL STAR(20,20),QL(10,20),RL(10,20)
C AUX. ARRAYS
        REAL WKAREA(130)
C MODE 3 ARRAYS
      REAL XX(10,10),VA(20),E(20),X(20),LRE(10),LIM(10),WJ(10)
      REAL W(10,10),V(10,10),VINV(10,10),F(10,10),AHAT(10,10)
        REAL A(10,10),B(10,10),C(10,10)
        COMMON/SYS/A,B,C,ZERO,IDGT,NS,NI,NO
        COMMON/AUG/F,AHAT/EIG/LRE,LIM/PAR/AL/GR/G
        COMMON/VEC/VA,E,X,WJ,W,XX,V,VINV
        COMMON/NSPA/ML,NL,NLC,PLC,MLC,STAR,QL,RL
        COMMON/AUX/AUX1,AUX2,AUX3
C       WRITE (5,1)                                     !**
C  1    FORMAT (1X,'SUBROUTINE GGRAD+++++++++++++++') !**
        CALL LINV2F (V,NS,10,VINV,IDGT,WKAREA,IER)
        CALL UERTST (IER,6HLINV2F)
C       CALL USWFM (12HMATRIX VINV:,12,VINV,10,NS,NS,4)   !**
        J=1
  10    CONTINUE
        IRS=202
        IU=J+20
        OPEN (ACCESS='RANDOM',RECORD SIZE=IRS,UNIT=IU
     1,MODE='BINARY',DEVICE='DSK',DISPOSE='SAVE')
C*************** Is Lambda-J real? ******************************
        READ (IU'1) LRE(J),LIM(J)
        IF (ABS(LIM(J)).GT.ABS(ZERO)) GO TO 30
C************* Find partials of J wrt elements of [XX], real case**
        READ (IU'3) ((NL(II,IJ),IJ=1,NI),II=1,NS)
        READ (IU'4) ((ML(II,IJ),IJ=1,NI),II=1,NI)
C       CALL USWFM (10HMATRIX NL:,10,NL,10,NS,NI,4)    !**
C       CALL USWFM (10HMATRIX ML:,10,ML,10,NI,NI,4)    !**
        GO TO 15
C*************** Find complex partials *************************
  30    IS=NS+NI
        NI2=2*NI
        NS2=2*NS
        INS=NS+1
        READ (IU'3) ((NLC(II,IJ),IJ=1,IS),II=1,NS)
        READ (IU'4) ((PLC(II,IJ),IJ=1,IS),II=1,NS)
        READ (IU'5) ((MLC(II,IJ),IJ=1,IS),II=1,NI)
        READ (IU'6) ((QL(II,IJ),IJ=1,NI2),II=1,NS)
        READ (IU'7) ((RL(II,IJ),IJ=1,NI2),II=1,NS)
```

```
C         CALL USWFM (11HMATRIX NLC:,11,NLC,10,NS,IS,4)      !**
C         CALL USWFM (11HMATRIX PLC:,11,PLC,10,NS,IS,4)      !**
C         CALL USWFM (11HMATRIX MLC:,11,MLC,10,NI,IS,4)      !**
C         CALL USWFM (10HMATRIX QL:,10,QL,10,NS,NI2,4)       !**
C         CALL USWFM (10HMATRIX RL:,10,RL,10,NS,NI2,4)       !**
     15   DO 100 I=1,NI
          G(I,J)=FLOAT(0)
          KJ=J
          KI=I
          CALL PFX(KI,KJ,IFLAG)
          IF (IFLAG.NE.0) GO TO 150
          DO 100 IP=1,NI
          DO 100 IQ=1,NS
          G(I,J)=G(I,J)+2*AL(IP,IQ)*F(IP,IQ)*AUX3(IP,IQ)
C         WRITE (5,2) IP,IQ,I,J,G(I,J)                          !**
C   2     FORMAT (20X,'PF',I2,I2,'/X',I2,I2,' =',E15.6,'PARTIAL SUMS')!**
    100   CONTINUE
          IF (IFLAG.EQ.1) J=J+1
          IF (J.GE.NS) GO TO 200
          J=J+1
          GO TO 10
    150   JD=J+1
          DO 70 IP=1,NI
          DO 70 IQ=1,NS
          G(I,J)=G(I,J)+2*AL(IP,IQ)*F(IP,IQ)*W(IP,IQ)
     70   CONTINUE
          G(I,JD)=FLOAT(0)
          DO 75 IP=1,NI
          DO 75 IQ=1,NS
          G(I,JD)=G(I,JD)+2*AL(IP,IQ)*F(IP,IQ)*AUX3(IP,IQ)
     75   CONTINUE
          GO TO 100
C************** Print [G], then find [G]/!!G!! *******
    200   CONTINUE
C         CALL USWFM (11HMATRIX [G]:,11,G,10,NI,NS,4)             !**
          CALL DBNORM(NI,NS)
          CALL USWFM (16HGradient matrix:,16,G,10,NI,NS,4)        !**
          RETURN
          END
```

```
C*************************************************************
C**************************************************************
      SUBROUTINE INSTEP
C-Function: Called by PFX calculates [AUX3]=([AUX1]-[AUX2])*[VINV]
C-IMSL routines called: UERTST,VMULFF,(USWFM).
C-Spectral Assignment routines: -
C-Logical devices; Input Unit:   -   Output Unit:   (5)
C-            Storage Unit(s): -
C-Random Access Files: -
      REAL AUX1(10,10),AUX2(10,10),AUX3(10,10)
      REAL XX(10,10),VA(20),E(20),X(20),WJ(10)
      REAL W(10,10),V(10,10),VINV(10,10),F(10,10),AHAT(10,10)
      REAL A(10,10),B(10,10),C(10,10)
      COMMON/VEC/VA,E,X,WJ,W,XX,V,VINV/AUG/F,AHAT
      COMMON/AUX/AUX1,AUX2,AUX3
      COMMON/SYS/A,B,C,ZERO,IDGT,NS,NI,NO
C     CALL USWFM (7H[AUX1]:,7,AUX1,10,NI,NS,4)       !**
C     CALL USWFM (7H[AUX2]:,7,AUX2,10,NS,NS,4)       !**
      CALL VMULFF(F,AUX2,NI,NS,NS,10,10,AUX3,10,IER)
      CALL UERTST(IER,6HVMULFF)
C     CALL USWFM (7H[AUX3]:,7,AUX3,10,NI,NS,4)       !**
      DO 10 II=1,NI
      DO 10 IJ=1,NS
      AUX2(II,IJ)=AUX1(II,IJ)-AUX3(II,IJ)
 10   CONTINUE
      CALL VMULFF(AUX2,VINV,NI,NS,NS,10,10,AUX3,10,IER)
      CALL UERTST(IER,6HVMULFF)
C     CALL USWFM (7H[AUX3]:,7,AUX3,10,NI,NS,4)       !**
      RETURN
      END
```

```
C*****************************************************************
C*****************************************************************
        SUBROUTINE DBNORM(NI,NS)
C-Function: Returns a normalized NIxNS matrix in itself.
C-IMSL routines called: -
C-Spectral Assignment routines: -
C-Logical devices; Input Unit:   -    Output Unit:   -
C-              Storage Unit(s): -
C-Random Access Files: -
        REAL G(10,10),NORM
        COMMON/GR/G
        NORM=FLOAT(0)
        DO 10 I=1,NI
        DO 10 J=1,NS
        NORM=NORM+G(I,J)**2
   10   CONTINUE
        NORM=SQRT(NORM)
        DO 20 I=1,NI
        DO 20 J=1,NS
        G(I,J)=G(I,J)/NORM
   20   CONTINUE
        RETURN
        END
```

```
C*********************************************************************
C*********************************************************************
        SUBROUTINE DESIGN
C-Function: Given a Designater matrix [X], calculates [V].
C-IMSL routines called: UERTST,VMULFF,(USWFM).
C-Spectral Assignment routines: -
C-Logical devices; Input Unit:   -    Output Unit:   (5)
C-             Storage Unit(s): IU=20+J
C-Random Access Files: FORxx.DAT where xx=20+J for J=1,NS.
C NULL SPACE ARRAYS
        REAL ML(10,10),NL(10,10)
        REAL NLC(10,20),PLC(10,20),MLC(10,20)
        REAL STAR(20,20),QL(10,20),RL(10,20)
C MODE 3 ARRAYS
      REAL XX(10,10),VA(20),E(20),X(20),LRE(10),LIM(10),WJ(10)
      REAL W(10,10),V(10,10),VINV(10,10)
        REAL A(10,10),B(10,10),C(10,10)
        COMMON/SYS/A,B,C,ZERO,IDGT,NS,NI,NO
        COMMON/EIG/LRE,LIM
        COMMON/VEC/VA,E,X,WJ,W,XX,V,VINV
        COMMON/NSPA/ML,NL,NLC,PLC,MLC,STAR,QL,RL
C       WRITE (5,1)                                       !**
C 1     FORMAT (1X,'SUBROUTINE DESIGN ++++++++++++++++++') !**
        IRS=202
        J=1
  10    IU=J+20
        OPEN (ACCESS='RANDOM',RECORD SIZE=IRS,UNIT=IU
     1,MODE='BINARY',DEVICE='DSK',DISPOSE='SAVE')
C*************** Is Lambda-J real? *****************************
        READ (IU'1) LRE(J),LIM(J)
        IF (ABS(LIM(J)).GT.ABS(ZERO)) GO TO 30
C*************** Find real VA=J-th column of [V] ***************
        READ (IU'3) ((NL(II,IJ),IJ=1,NI),II=1,NS)
C       CALL USWFM (10HMATRIX NL:,10,NL,10,NS,NI,4)     !**
        DO 20 IV=1,NI
        X(IV)=XX(IV,J)
  20    CONTINUE
C******** Find VA=[NL]*X and put it in J-th column of [V] ******
        CALL VMULFF (NL,X,NS,NI,1,10,20,VA,20,IER)
        CALL UERTST (IER,6HVMULFF)
C       CALL USWFV (10HVECTOR VA:,10,VA,NS,1,4)      !**
        DO 25 IV=1,NS
        V(IV,J)=VA(IV)
  25    CONTINUE
  29    IF (J.GE.NS) GO TO 100
        J=J+1
        GO TO 10
C**************** Find complex VA's ****************************
  30    INI=NI+1
        NI2=2*NI
        NS2=2*NS
        INS=NS+1
        READ (IU'6) ((QL(II,IJ),IJ=1,NI2),II=1,NS)
        READ (IU'7) ((RL(II,IJ),IJ=1,NI2),II=1,NS)
C       CALL USWFM (10HMATRIX QL:,10,QL,10,NS,NI2,4)    !**
C       CALL USWFM (10HMATRIX RL:,10,RL,10,NS,NI2,4)    !**
```

```
             IC=J+1
      C****** Form [STAR] and double length X ***********************
             DO 135 II=1,NS
             DO 135 IJ=1,NI2
             STAR(II,IJ)=QL(II,IJ)
       135   CONTINUE
             DO 140 II=INS,NS2
             DO 140 IJ=1,NI2
             IDUM=II-NS
             STAR(II,IJ)=RL(IDUM,IJ)
       140   CONTINUE
      C      CALL USWFM (12HMATRIX STAR:,12,STAR,20,NS2,NI2,4) !**
             DO 40 IV=1,NI
             X(IV)=XX(IV,J)
        40   CONTINUE
             DO 50 IV=INI,NI2
             IVDUM=IV-NI
             X(IV)=XX(IVDUM,IC)
        50   CONTINUE
      C      CALL USWFV (10HVECTOR XT:,10,X,NI2,1,4)                !**
      C***** Find VA=[*].X and partition it to [V]j, [V]j+1 **********
             CALL VMULFF (STAR,X,NS2,NI2,1,20,20,VA,20,IER)
             CALL UERTST (IER,6HVMULFF)
      C      CALL USWFV (10HVECTOR VA:,10,VA,NS2,1,4)             !**
             DO 60 IV=1,NS
             V(IV,J)=VA(IV)
             IVD=IV+NS
             V(IV,IC)=VA(IVD)
        60   CONTINUE
             J=IC
             GO TO 29
      C************** Print [V]   ********************************
       100   CONTINUE
      C      CALL USWFM (11HMATRIX [V]:,11,V,10,NS,NS,4)            !**
      C      WRITE (5,2)                                  !**
      C   2  FORMAT (1X,'EXITING DESIGN =============') !**
             RETURN
             END
```

```
C****************************************************************
C****************************************************************
           SUBROUTINE PFX(I,J,IFLAG)
C-Function: Returns p[F]/p[x]ij .
C-IMSL routines called: UERTST,LLSQF,VMULFF,(USWFM).
C-Spectral Assignment routines: INSTEP.
C-Logical devices; Input Unit:   -    Output Unit:   (5)
C-             Storage Unit(s): -
C-Random Access Files: -
C GRADIENT ARRAYAS
           REAL AUX1(10,10),AUX2(10,10),AUX3(10,10)
C NULL SPACE APRAYS
           REAL ML(10,10),NL(10,10),ALPHA(20,20),BETA(20,20)
           REAL NLC(10,20),PLC(10,20),MLC(10,20)
           REAL STAR(20,20),QL(10,20),RL(10,20)
C AUX. ARRAYS
           REAL WKAREA(130),H(20)
C MODE 3 ARRAYS
         REAL XX(10,10),VA(20),E(20),X(20),LRE(10),LIM(10),WJ(10)
         REAL W(10,10),V(10,1C),VINV(10,10),F(10,10),AHAT(10,10)
           INTEGER IPA(20)
           REAL A(10,10),B(10,10),C(10,10)
           COMMON/SYS/A,B,C,ZERO,IDGT,NS,NI,NO
           COMMON/EIG/LRE,LIM
           COMMON/VEC/VA,E,X,WJ,W,XX,V,VINV
           COMMON/NSPA/ML,NL,NLC,PLC,MLC,STAR,QL,RL
           COMMON/AUX/AUX1,AUX2,AUX3
C**************** Is Lambda-J real? ******************************
           IF (ABS(LIM(J)).GT.ABS(ZERO)) GO TO 30
C************* Find partials of J wrt elements of [XX], real case**
           DO 15 II=1,NI
           DO 15 IJ=1,NS
           AUX1(II,IJ)=FLOAT(0)
           IF (IJ.EQ.J) AUX1(II,IJ)=-ML(II,I)
  15       CONTINUE
           DO 20 II=1,NS
           DO 20 IJ=1,NS
           AUX2(II,IJ)=FLOAT(0)
           IF (IJ.EQ.J) AUX2(II,IJ)=NL(II,I)
  20       CONTINUE
C          CALL USWFM (7H[AUX1]:,7,AUX1,10,NI,NS,4)       !**
C          CALL USWFM (7H[AUX2]:,7,AUX2,10,NS,NS,4)       !**
           CALL INSTEP
           IFLAG=0
           GO TO 999
C**************** Find complex partials *************************
  30       IS=NS+NI
           NI2=2*NI
           NS2=2*NS
           INS=NS+1
           JD=J
           JC=J+1
           INOW=I
C***** FORM [STAR], [ALPHA],[BETA] ******************************
           DO 110 II=1,NS
           DO 110 IJ=1,NI2
           STAR(II,IJ)=QL(II,IJ)
           IDUM=II+NS
           STAR(IDUM,IJ)=RL(II,IJ)
 110       CONTINUE
```

```
C          CALL USWFM (12HMATRIX STAR:,12,STAR,20,NS2,NI2,4)   !**
           DO 135 II=1,NS
           DO 135 IJ=1,IS
           ALPHA(II,IJ)=NLC(II,IJ)
  135      CONTINUE
           DO 140 II=INS,NS2
           DO 140 IJ=1,IS
           IDUM=II-NS
           ALPHA(II,IJ)=-PLC(IDUM,IJ)
  140      CONTINUE
C          CALL USWFM (14HMATRIX ALPHA :,14,ALPHA,20,NS2,IS,4) !**
           DO 185 II=1,NS
           DO 185 IJ=1,IS
           BETA(II,IJ)=PLC(II,IJ)
  185      CONTINUE
           DO 190 II=INS,NS2
           DO 190 IJ=1,IS
           IDUM=II-NS
           BETA(II,IJ)=NLC(IDUM,IJ)
  190      CONTINUE
C           CALL USWFM (13HMATRIX BETA :,13,BETA,20,NS2,IS,4)   !**
   50      CONTINUE
           DO 55 II=1,NS2
           E(II)=STAR(II,INOW)
           VA(II)=STAR(II,INOW)
   55      CONTINUE
C          CALL USWFV (19HI-th column of [*]:,19,E,NS2,1,4)    !**
C          CALL USWFV (19HI-th column of [*]:,19,VA,NS2,1,4)    !**
           CALL LLSQF (ALPHA,20,NS2,IS,E,-1.0,IS,X,H,IPA,IER)
           CALL UERTST (IER,6HLLSQF )
C          CALL USWFV (15HVECTOR [TM1]-I:,15,X,IS,1,4)         !**
C*** Form E=[MLC]*[TM1]i and put E in J-th column of [AUX1] *****
           CALL VMULFF (MLC,X,NI,IS,1,10,20,E,20,IER)
           CALL UERTST (IER,6HVMULFF)
C          CALL USWFV (10H[AUX1]-J :,10,E,NI,1,4)         !**
           DO 180 IV=1,IS
           X(IV)=FLOAT(0)
  180      CONTINUE
           CALL LLSQF (BETA,20,NS2,IS,VA,-1.0,IS,X,H,IPA,IER)
           CALL UERTST (IER,6HLLSQF )
C          CALL USWFV (15HVECTOR [TM2]-I:,15,X,IS,1,4)         !**
C** Form VA=[MLC]*[TM1]i and put VA in J+1th column of [AUX1] ***
           CALL VMULFF (MLC,X,NI,IS,1,10,20,VA,20,IER)
           CALL UERTST (IER,6HVMULFF)
C          CALL USWFV (10H[AUX1]J+1:,10,VA,NI,1,4)         !**
           DO 60 II=1,NI
           DO 60 IJ=1,NS
           AUX1(II,IJ)=FLOAT(0)
           IF (IJ.EQ.JD) AUX1(II,IJ)=E(II)
           IF (IJ.EQ.JC) AUX1(II,IJ)=VA(II)
   60      CONTINUE
C          CALL USWFM (8H[AUX1] :,8,AUX1,10,NI,NS,4)      !**
           DO 70 II=1,NS
           DO 70 IJ=1,NS
           AUX2(II,IJ)=FLOAT(0)
           IF (IJ.EQ.JD) AUX2(II,IJ)=QL(II,INOW)
           IF (IJ.EQ.JC) AUX2(II,IJ)=RL(II,INOW)
   70      CONTINUE
C          CALL USWFM (8H[AUX2] :,8,AUX2,10,NS,NS,4)       !**
           CALL INSTEP
           IF (INOW.NE.I) GO TO 999
```

```
      IFLAG=1
      DO 80 II=1,NI
      DO 80 IJ=1,NS
      W(II,IJ)=AUX3(II,IJ)
   80 CONTINUE
      INOW=I+NI
      GO TO 50
  999 RETURN
      END
```

```
C*****************************************************************

C*****************************************************************
        SUBROUTINE MODE7
C-Function: Main routine for Sensitivity Reduction.
C-IMSL routines called: UERSET,UERTST,LINV2F,USWFM.
C-Spectral Assignment routines: SEARCH,TRAN,SGRAD,SCOST,SENS.
C-Logical devices; Input Unit:   5    Output Unit:   5
C-             Storage Unit(s): IU=20,IUT=20+NS+1,IU=20+J for J=1,NS.
C-Random Access Files: SYSTEM.DAT,CURRNT.DAT,FORxx.DAT where xx=20+J.
        REAL WKAREA(130),U(10,10)
C GRADIENT ARRAYAS
        REAL L(10),P(10),DAD(10,10),DBD(10,10),DAHD(10,10)
C NULL SPACE ARRAYS
        REAL ML(10,10),NL(10,10)
        REAL NLC(10,20),PLC(10,20),MLC(10,20)
        REAL STAR(20,20),QL(10,20),RL(10,20)
C MODE 3 ARRAYS
      REAL XX(10,10),VA(20),E(20),X(20),LRE(10),LIM(10),WJ(10)
      REAL W(10,10),V(10,10),VINV(10,10),F(10,10),AHAT(10,10)
        REAL A(10,10),B(10,10),C(10,10)
        COMMON/SYS/A,B,C,ZERO,IDGT,NS,NI,NO
      COMMON/AUG/F,AHAT/EIG/LRE,LIM/WET/L,P/GR/G/SEN/DAD,DBD,DAHD/LEG/U
        COMMON/VEC/VA,E,X,WJ,W,XX,V,VINV
        COMMON/NSPA/ML,NL,NLC,PLC,MLC,STAR,QL,RL
        EXTERNAL SCOST,SGRAD
        CALL UERSET(3,LEVOLD)
        IU=20
        READ (IU'1) NS,NI,NO,IDGT,ZERO
        READ (IU'2) ((A(II,IJ),IJ=1,NS),II=1,NS)
        READ (IU'3) ((B(II,IJ),IJ=1,NI),II=1,NS)
        IUT=20+NS+1
      OPEN (FILE='CURRNT.DAT',ACCESS='RANDOM',RECORD SIZE=102
     1,UNIT=IUT,MODE='BINARY',DEVICE='DSK',DISPOSE='SAVE')
        READ (IUT'1) ((V(II,IJ),IJ=1,NS),II=1,NS)
        READ (IUT'2) ((XX(II,IJ),IJ=1,NS),II=1,NI)
        READ (IUT'4) ((F(II,IJ),IJ=1,NS),II=1,NI)
        READ (IUT'5) ((AHAT(II,IJ),IJ=1,NS),II=1,NS)
        CALL USWFM (10HMATRIX V :,10,V,10,NS,NS,4)          !**
        CALL USWFM (10HMATRIX XX:,10,XX,10,NI,NS,4)         !**
        CALL USWFM (14HGAIN MATRIX F:,14,F,10,NI,NS,4)      !**
        CALL USWFM (12HMATRIX AHAT:,12,AHAT,10,NS,NS,4)     !**
        DO 30 J=1,NS
        IU=J+20
        IRS=202
        OPEN (ACCESS='RANDOM',RECORD SIZE=IRS,UNIT=IU
     1,MODE='BINARY',DEVICE='DSK',DISPOSE='SAVE')
        READ (IU'1) LRE(J),LIM(J)
   30   CONTINUE
        WRITE (5,1)
    1   FORMAT (1X,20(1H*),30H MODE 7:SENSITIVITY REDUCTION ,20(1H*),//
     1,1X,23H Set weighting factors:,/
     2,1x,34H Eigenvalue weighting factors are:)
        DO 10 IV=1,NS
        L(IV)=FLOAT(1)
        WRITE (5,3) IV,L(IV)
    3   FORMAT (1X,2HL(,I2,2H)=,F15.6)
   10   CONTINUE
```

```
      WRITE (5,5)
    5 FORMAT (1X,15HWish to change?)
      READ (5,*) KL
      IF (KL.LE.0) GO TO 11
      READ (5,*) (L(IV),IV=1,NS)
   11 WRITE (5,2)
    2 FORMAT (1X,34HEigenvector weighting factors are:)
      DO 15 IV=1,NS
      P(IV)=FLOAT(1)
      WRITE (5,4) IV,P(IV)
    4 FORMAT (1X,2HM(,I2,2H)=,F15.6)
   15 CONTINUE
      WRITE (5,5)
      READ (5,*) KK
      IF (KK.LE.0) GO TO 20
      READ (5,*) (P(IV),IV=1,NS)
   20 WRITE (5,6)
    6 FORMAT (1X,14HEnter [dA/dP]:)
      READ (5,*) ((DAD(II,IJ),IJ=1,NS),II=1,NS)
      WRITE (5,7)
    7 FORMAT (1X,14HEnter [dB/dP]:)
      READ (5,*) ((DBD(II,IJ),IJ=1,NI),II=1,NS)
      CALL SENS
      IDG=IDGT
      CALL LINV2F (V,NS,10,U,IDG,WKAREA,IER)
      CALL UERTST(IER,6HLINV2F)
C     CALL USWFM (10HMATRIX7UT:,10,U,10,NS,NS,4)        !**
      CALL tran(U,NS,NS)
C     CALL USWFM (10HMATRIX7U :,10,U,10,NS,NS,4)        !**
      CALL SCOST(CJ)
      CALL SGRAD
      CALL SEARCH(CJ,SCOST,SGRAD,7)
      WRITE (IUT'1) ((V(II,IJ),IJ=1,NS),II=1,NS)
      WRITE (IUT'2) ((XX(II,IJ),IJ=1,NS),II=1,NI)
      WRITE (IUT'4) ((F(II,IJ),IJ=1,NS),II=1,NI)
      WRITE (IUT'5) ((AHAT(II,IJ),IJ=1,NS),II=1,NS)
      CALL USWFM (10HMATRIX V :,10,V,10,NS,NS,4)        !**
      WRITE (5,902)
  902 FORMAT (1X,44HWISH TO DISPLAY THE NORMALIZED EIGENVECTORS?)
      READ (5,*) KS
      IF (KS.LE.0) GO TO 903
      CALL DSPLAY (NS,ZERO)
  903 CONTINUE
      CALL USWFM (10HMATRIX XX:,10,XX,10,NI,NS,4)       !**
      CALL USWFM (14HGAIN MATRIX F:,14,F,10,NI,NS,4)    !**
      CALL USWFM (12HMATRIX AHAT:,12,AHAT,10,NS,NS,4)   !**
      RETURN
      END
```

```
C*****************************************************************
C*****************************************************************
            SUBROUTINE SCOST(CJ)
C-Function: Calculates the COST function for Sensitivity Reduction.
C-IMSL routines called: (USWFM).
C-Spectral Assignment routines: ZK, and Function routine T .
C-Logical devices; Input Unit:   -    Output Unit:   5
C-             Storage Unit(s): -
C-Random Access Files: -
            REAL V(10,10),U(10,10),L(10),P(10)
            REAL VJ(10),VJ1(10),UJ(10),UJ1(10),LRE(10),LIM(10)
         REAL XX(10,10),VA(20),E(20),X(20),WJ(10)
         REAL W(10,10),VINV(10,10)
            REAL A(10,10),B(10,10),C(10,10)
            INTEGER Q
            COMMON/SYS/A,B,C,ZERO,IDGT,NS,NI,NO
            COMMON/EIG/LRE,LIM/WET/L,P/LEG/U
            COMMON/VEC/VA,E,X,WJ,W,XX,V,VINV
            CJ1=FLOAT(0)
            CJ2=FLOAT(0)
            DO 100 J=1,NS
            JC=J+1
            RELJ=LRE(J)
            XIMJ=LIM(J)
c           WRITE (5,1) LRE(J),RELJ,LIM(J),XIMJ        !**
c   1       FORMAT (1X,'LRE(J)=RELJ',2F15.6,'LIM(J)=XIMJ',2F15.6) !**
            DO 10 IV=1,NS
            VJ1(IV)=FLOAT(0)
            UJ1(IV)=FLOAT(0)
            VJ(IV)=V(IV,J)
            UJ(IV)=U(IV,J)
            IF (ABS(XIMJ).LE.ABS(ZERO)) GO TO 10
            VJ1(IV)=V(IV,JC)
            UJ1(IV)=U(IV,JC)
   10       CONTINUE
c           CALL USWFV(11HVECTOR  VJ:,11, VJ,NS,1,4)      !**
c           CALL USWFV(11HVECTOR VJ1:,11,VJ1,NS,1,4)      !**
c           CALL USWFV(11HVECTOR  UJ:,11, UJ,NS,1,4)      !**
c           CALL USWFV(11HVECTOR UJ1:,11,UJ1,NS,1,4)      !**
            IF (ABS(XIMJ).LE.ABS(ZERO)) GO TO 20
         CJ1=CJ1+L(J)*((T(1,VJ,UJ)-T(1,VJ1,UJ1))**2+(T(1,VJ1,UJ)+
        1T(1,VJ,UJ1))**2)
            GO TO 30
   20       CJ1=CJ1+L(J)*T(1,VJ,UJ)**2
   30       SUM=FLOAT(0)
            DO 50 IQ=1,NS
            Q=IQ
            NJ=J
            CALL ZK(Q,NJ,RELJ,XIMJ,ZRE,ZIM)
            SUM=SUM+ZRE**2+ZIM**2
   50       CONTINUE
            CJ2=CJ2+P(J)*SUM
  100       CONTINUE
            CJ=CJ1+CJ2
            WRITE (5,2) CJ1,CJ2
    2       FORMAT (1X,3HJ1=,F15.6,5X,3HJ2=,F15.6)
            RETURN
            END
```

```
C*************************************************************
C*************************************************************
            SUBROUTINE SGRAD
C-Function: Calculates the Gradient for Sensitivity Reduction.
C-IMSL routines called: UERTST,LINV2F,VMULFF,USWFM.
C-Spectral Assignment routines: PU,DBNORM,PFX,ZK,FRAC,Function T.
C-Logical devices; Input Unit:   -    Output Unit:   5
C-           Storage Unit(s): IU=20+J for J=1,NS.
C-Random Access Files: FORxx.dat where xx=20+J.
C GRADIENT ARRAYAS
            REAL G(10,10),U(10,10),PJ1(10,10),PJ2(10,10),L(10),P(10)
            REAL VJ(10),VJ1(10),PVJX(10),PVJX1(10),PVJ1X(10),PVJ1X1(10)
            REAL UJ(10),UJ1(10),PUKX(10),PUKX1(10),PUK1X(10),PUK1X1(10)
            REAL VM(10),VM1(10),UK(10),UK1(10)
            REAL PUR(10,10),PUC(10,10),PUC1(10,10)
C NULL SPACE ARRAYS
            REAL ML(10,10),NL(10,10)
            REAL NLC(10,20),PLC(10,20),MLC(10,20)
            REAL STAR(20,20),QL(10,20),RL(10,20)
C AUX. ARRAYS
            REAL WKAREA(130)
C MODE 3 ARRAYS
         REAL XX(10,10),VA(20),E(20),X(20),LRE(10),LIM(10),WJ(10)
         REAL W(10,10),V(10,10),VINV(10,10)
            REAL AUX1(10,10),AUX2(10,10),AUX3(10,10),AUX4(10,10)
            REAL DAD(10,10),DBD(10,10),DAHD(10,10)
            INTEGER Q,FLK,FLJ,FLM
            REAL A(10,10),B(10,10),C(10,10)
            COMMON/SYS/A,B,C,ZERO,IDGT,NS,NI,NO
            COMMON/EIG/LRE,LIM/GR/G/LEG/U/WET/L,P/PJ/PJ1,PJ2
            COMMON/VEC/VA,E,X,WJ,W,XX,V,VINV
            COMMON/NSPA/ML,NL,NLC,PLC,MLC,STAR,QL,RL
            COMMON/AUX/AUX1,AUX2,AUX3/SEN/DAD,DBD,DAHD/AAUX/AUX4
            IGT=IDGT
            CALL LINV2F (V,NS,10,VINV,IGT,WKAREA,IER)
            CALL UERTST (IER,6HLINV2F)
C           CALL USWFM (12HMATRIX VINV:,12,VINV,10,NS,NS,4)    !**
            IRS=202
            J=1
   10       FLJ=0
            IU=J+20
            RELJ=LRE(J)
            XIMJ=LIM(J)
            IF (ABS(LIM(J)).GT.ABS(ZERO)) FLJ=1
            IF (FLJ.EQ.1) GO TO 12
            DO 11 IV=1,NS
            VJ(IV)=V(IV,J)
            UJ(IV)=U(IV,J)
            VJ1(IV)=FLOAT(0)
            UJ1(IV)=FLOAT(0)
   11       CONTINUE
            READ (IU'3) ((NL(II,IJ),IJ=1,NI),II=1,NS)
            READ (IU'4) ((ML(II,IJ),IJ=1,NI),II=1,NI)
C           CALL USWFM (10HMATRIX ML:,10,ML,10,NI,NI,4)       !**
C           CALL USWFM (10HMATRIX NL:,10,NL,10,NS,NI,4)       !**
            GO TO 14
```

C-2

```
 12     NI2=2*NI
        IS=NS+NI
        READ (IU'3) ((NLC(II,IJ),IJ=1,IS),II=1,NS)
        READ (IU'4) ((PLC(II,IJ),IJ=1,IS),II=1,NS)
        READ (IU'5) ((MLC(II,IJ),IJ=1,IS),II=1,NI)
        READ (IU'6) ((QL(II,IJ),IJ=1,NI2),II=1,NS)
        READ (IU'7) ((RL(II,IJ),IJ=1,NI2),II=1,NS)
        JC=J+1
        DO 13 IV=1,NS
        VJ1(IV)=V(IV,JC)
        UJ1(IV)=U(IV,JC)
 13     CONTINUE
 14     DO 110 I=1,NI
        KI=I
        KJ=J
        CALL PFX(KI,KJ,IFLAG)
        CALL VMULFF(DBD,AUX3,NS,NI,NS,10,10,AUX4,10,IER)
        CALL UERTST (IER,6HVMULFF)
C       CALL USWFM (12HMATRIX AUX4:,12,AUX4,10,NS,NS,4)     !**
        IF (FLJ.NE.1) GO TO 115
        CALL VMULFF(DBD,W,NS,NI,NS,10,10,AUX4,10,IER)
        CALL VMULFF(DBD,AUX3,NS,NI,NS,10,10,W,10,IER)
        CALL UERTST (IER,6HVMULFF)
        DO 114 II=1,NS
        DO 114 IJ=1,NS
        AUX3(II,IJ)=W(II,IJ)
114     CONTINUE
C       CALL USWFM (12HMATRIX AUX4:,12,AUX4,10,NS,NS,4)     !**
C       CALL USWFM (12HMATRIX AUX3:,12,AUX3,10,NS,NS,4)     !**
115     IF (FLJ.EQ.1) GO TO 16
        DO 15 IV=1,NS
        PVJX(IV)=NL(IV,I)
        DO 15 IW=1,NS
        W(IV,IW)=FLOAT(0)
        IF (IW.EQ.J) W(IV,IW)=NL(IV,I)
 15     CONTINUE
        CALL PU(PUR)
        TERM=FLOAT(0)
        FLM=0
        DO 116 M=1,NS
        IF (ABS(LIM(M)).GT.ABS(ZERO)) FLM=1
        MC=M+1
        DO 216 IV=1,NS
        VM(IV)=V(IV,M)
        UK(IV)=U(IV,M)
        PUKX(IV)=PUR(IV,M)
        VM1(IV)=FLOAT(0)
        UK1(IV)=FLOAT(0)
        PUK1X(IV)=FLOAT(0)
        IF (FLM.NE.1) GO TO 216
        VM1(IV)=V(IV,MC)
        UK1(IV)=U(IV,MC)
        PUK1X(IV)=PUR(IV,MC)
216     CONTINUE
        IF(FLM.EQ.1) GO TO 316
        TERM=TERM+L(M)*(T(2,VM,UK)+T(1,VM,PUKX))*T(1,VM,UK)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      GO TO 116
  316 TERM=TERM+L(M)*((T(2,VM,UK)-T(2,VM1,UK1)+T(1,VM,PUKX)-T(1,VM1,PUK1
     1X))*(T(1,VM,UK)-T(1,VM1,UK1))+(T(2,VM,UK1)+T(2,VM1,UK)+T(1,VM,PUK1
     2X)+T(1,VM1,PUKX))*(T(1,VM1,UK)+T(1,VM,UK1)))
C     WRITE (6,*) TERM                                        !**
  116 CONTINUE
      PJ1(I,J)=2*(L(J)*T(1,PVJX,UJ)*T(1,VJ,UJ)+TERM)
C     WRITE (5,201)  I,J,PJ1(I,J)                             !**
C 201 FORMAT(1X,'GGGGGG I=',I2,'J=',I2,'PJ1(I,J)=',F15.6) !**
      GO TO 20
   16 INI=I+NI
      DO 17 IV=1,NS
      PVJX(IV)=QL(IV,I)
      PVJX1(IV)=QL(IV,INI)
      PVJ1X(IV)=RL(IV,I)
      PVJ1X1(IV)=RL(IV,INI)
      DO 17 IW=1,NS
      W(IV,IW)=FLOAT(0)
      IF (IW.EQ.J) W(IV,IW)=QL(IV,I)
      IF (IW.EQ.JC) W(IV,IW)=RL(IV,I)
   17 CONTINUE
      CALL PU(PUC)
      TERM=FLOAT(0)
      FLM=0
      DO 117 M=1,NS
      IF (FLM.EQ.1) GO TO 317
      FLM=0
      MC=M+1
      IF (ABS(LIM(M)).GT.ABS(ZERO)) FLM=1
      DO 217 IV=1,NS
      VM(IV)=V(IV,M)
      UK(IV)=U(IV,M)
      PUKX(IV)=PUC(IV,M)
      VM1(IV)=FLOAT(0)
      UK1(IV)=FLOAT(0)
      PUK1X(IV)=FLOAT(0)
      IF (FLM.NE.1) GO TO 217
      VM1(IV)=V(IV,MC)
      UK1(IV)=U(IV,MC)
      PUK1X(IV)=PUC(IV,MC)
  217 CONTINUE
     IF(FLM.NE.1) TERM=TERM+L(M)*(T(2,VM,UK)+T(1,VM,PUKX))*T(1,VM,UK)
      IF (FLM.NE.1) GO TO 117
     TERM=TERM+L(M)*((T(2,VM,UK)-T(2,VM1,UK1)+T(1,VM,PUKX)-T(1,VM1
     1,PUK1X))*(T(1,VM,UK)-T(1,VM1,UK1))+(T(2,VM,UK1)+T(2,VM1,UK)
     2+T(1,VM,PUK1X)+T(1,VM1,PUKX))*(T(1,VM1,UK)+T(1,VM,UK1)))
      GO TO 117
  317 FLM=0
  117 CONTINUE
     PJ1(I,J)=2*(L(J)*((T(1,PVJX,UJ)-T(1,PVJ1X,UJ1))*(T(1,VJ,UJ)-
     1T(1,VJ1,UJ1))+(T(1,PVJX,UJ1)+T(1,PVJ1X,UJ))*(T(1,VJ1,UJ)+
     2T(1,VJ,UJ1)))+TERM)
      DO 18 IV=1,NS
      DO 18 IW=1,NS
      W(IV,IW)=FLOAT(0)
      IF (IW.EQ.J) W(IV,IW)=QL(IV,INI)
```

```
      IF (IW.EQ.JC) W(IV,IW)=RL(IV,INI)
  18  CONTINUE
      CALL PU(PUC1)
      TERM=FLOAT(0)
      FLM=0
      DO 118 M=1,NS
      IF (FLM.EQ.1) GO TO 318
      FLM=0
      MC=M+1
      IF (ABS(LIM(M)).GT.ABS(ZERO)) FLM=1
      DO 218 IV=1,NS
      VM(IV)=V(IV,M)
      UK(IV)=U(IV,M)
      PUKX1(IV)=PUC1(IV,M)
      VM1(IV)=FLOAT(0)
      UK1(IV)=FLOAT(0)
      PUK1X1(IV)=FLOAT(0)
      IF (FLM.NE.1) GO TO 218
      VM1(IV)=V(IV,MC)
      UK1(IV)=U(IV,MC)
      PUK1X1(IV)=PUC1(IV,MC)
 218  CONTINUE
     IF(FLM.NE.1) TERM=TERM+L(M)*(T(3,VM,UK)+T(1,VM,PUKX1))*T(1,VM,UK)
      IF (FLM.NE.1) GO TO 118
     TERM=TERM+L(M)*((T(3,VM,UK)-T(3,VM1,UK1)+T(1,VM,PUKX1)-T(1,VM1
    1,PUK1X1))*(T(1,VM,UK)-T(1,VM1,UK1))+(T(3,VM,UK1)+T(3,VM1,UK)
    2+T(1,VM,PUK1X1)+T(1,VM1,PUKX1))*(T(1,VM1,UK)+T(1,VM,UK1)))
      GO TO 118
 318  FLM=0
 118  CONTINUE
     PJ1(I,JC)=2*(L(JC)*((T(1,PVJX1,UJ)-T(1,PVJ1X1,UJ1))*(T(1,VJ,UJ)-
    1T(1,VJ1,UJ1))+(T(1,PVJX1,UJ1)+T(1,PVJ1X1,UJ))*(T(1,VJ1,UJ)+
    2T(1,VJ,UJ1)))+TERM)
      PJ2(I,JC)=FLOAT(0)
  20  PJ2(I,J)=FLOAT(0)
      M=1
  40  FLM=0
      MC=M+1
      IF (ABS(LIM(M)).GT.ABS(ZERO)) FLM=1
      DO 45 IV=1,NS
      VM(IV)=V(IV,M)
      VM1(IV)=FLOAT(0)
      IF (FLM.EQ.1) VM1(IV)=V(IV,MC)
  45  CONTINUE
      TERM1=FLOAT(0)
      TIM1=FLOAT(0)
      TERM11=FLOAT(0)
      TIM11=FLOAT(0)
      SUMQ1=FLOAT(0)
      SUMQ2=FLOAT(0)
      DO 80 Q=1,NS
      KQ=Q
      IF (M.EQ.J) GO TO 50
      VQJ=V(Q,J)
      VQJ1=FLOAT(0)
      IF (FLJ.EQ.1) VQJ1=V(Q,JC)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      IF (FLJ.EQ.1) GO TO 48
      PQJX=NL(Q,I)
      PQJ1X=FLOAT(0)
      GO TO 49
 48   PQJX=QL(Q,I)
      PQJ1X=RL(Q,I)
      PQJX1=QL(Q,INI)
      PQJ1X1=RL(Q,INI)
 49   CALL FRAC(KQ,KJ,M,LRE(M),LIM(M),1,UJ,UJ1,VM,VM1,PQJX,PQJ1X,
     1TERM1,TIM1)
C     WRITE (5,202) TERM1,FRE,TIM1,FIM,J,I,M,Q          !**
C 202 FORMAT (1X,'TERM1,FRE,TIM1,FIM',4F15.6,/,1X,'J,I,M,Q',4I2) !**
      IF (FLJ.NE.1) GO TO 50
      CALL FRAC(KQ,KJ,M,LRE(M),LIM(M),1,UJ,UJ1,VM,VM1,PQJX1,PQJ1X1,
     1TERM11,TIM11)
 50   CONTINUE
      TERM2=FLOAT(0)
      TIM2=FLOAT(0)
      TERM21=FLOAT(0)
      TIM21=FLOAT(0)
      K=1
 51   FLK=0
      KC=K+1
      IF (ABS(LIM(K)).GT.ABS(ZERO)) FLK=1
      IF (FLK.NE.1.AND.K.EQ.M) GO TO 58
      IF (FLK.EQ.1.AND.K.EQ.M) GO TO 57
      DO 52 IV=1,NS
      IF (FLJ.NE.1) PUKX(IV)=PUR(IV,K)
      IF (FLJ.EQ.1) PUKX(IV)=PUC(IV,K)
      PUK1X(IV)=FLOAT(0)
      IF (FLK.EQ.1.AND.FLJ.NE.1) PUK1X(IV)=PUR(IV,KC)
      IF (FLK.EQ.1.AND.FLJ.EQ.1) PUK1X(IV)=PUC(IV,KC)
      PUKX1(IV)=FLOAT(0)
      IF (FLJ.EQ.1) PUKX1(IV)=PUC1(IV,K)
      PUK1X1(IV)=FLOAT(0)
      IF (FLK.EQ.1.AND.FLJ.EQ.1) PUK1X1(IV)=PUC1(IV,KC)
      UK(IV)=U(IV,K)
      UK1(IV)=FLOAT(0)
      IF (FLK.EQ.1) UK1(IV)=U(IV,KC)
 52   CONTINUE
      VQK=V(Q,K)
      VQK1=FLOAT(0)
      IF (FLK.EQ.1) VQK1=V(Q,KC)
      CALL FRAC(KQ,K,M,LRE(M),LIM(M),1,PUKX,PUK1X,VM,VM1,VQK,VQK1,
     1FRE1,FIM1)
     CALL FRAC(KQ,K,M,LRE(M),LIM(M),2,UK,UK1,VM,VM1,VQK,VQK1,FRE,FIM)
C     WRITE (5,204) TERM2,FRE,TIM2,FIM,J,I,M,Q,K        !**
C 204 FORMAT (1X,'TERM2,FRE,TIM2,FIM',4F15.6,/,1X,'J,I,M,Q,K',5I2) !**
      TERM2=TERM2+FRE+FRE1
      TIM2=TIM2+FIM+FIM1
C     WRITE (5,204) TERM2,FRE,TIM2,FIM,J,I,M,Q,K        !**
      IF (FLJ.NE.1) GO TO 57
      CALL FRAC(KQ,K,M,LRE(M),LIM(M),1,PUKX1,PUK1X1,VM,VM1,VQK,
     1VQK1,FRE1,FIM1)
     CALL FRAC(KQ,K,M,LRE(M),LIM(M),3,UK,UK1,VM,VM1,VQK,VQK1,FRE,FIM)
      TERM21=TERM21+FRE+FRE1
```

```
            TIM21=TIM21+FIM+FIM1
      57    IF (FLK.EQ.1) K=K+1
      58    IF (K.EQ.NS) GO TO 60
            K=K+1
            GO TO 51
      60    CONTINUE
            TERM3=FLOAT(0)
            TIM3=FLOAT(0)
            TERM31=FLOAT(0)
            TIM31=FLOAT(0)
            IF (M.NE.J) GO TO 70
            K=1
      61    FLK=0
            KC=K+1
            IF (ABS(LIM(K)).GT.ABS(ZERO)) FLK=1
            IF (FLK.NE.1.AND.K.EQ.J) GO TO 68
            IF (FLK.EQ.1.AND.K.EQ.J) GO TO 67
            DO 62 IV=1,NS
            UK(IV)=U(IV,K)
            UK1(IV)=FLOAT(0)
            IF (FLK.EQ.1) UK1(IV)=U(IV,KC)
      62    CONTINUE
            VQK=V(Q,K)
            VQK1=FLOAT(0)
            IF (FLK.EQ.1) VQK1=V(Q,KC)
          CALL FRAC(KQ,K,J,LRE(J),LIM(J),1,UK,UK1,PVJX,PVJ1X,VQK,VQK1,FRE
         1,FIM)
C           WRITE (5,205) TERM3,FRE,TIM3,FIM,J,I,M,Q,K          !**
C 205       FORMAT (1X,'TERM3,FRE,TIM3,FIM',4F15.6,/,1X,'J,I,M,Q,K',5I2) !**
            TERM3=TERM3+FRE
            TIM3=TIM3+FIM
C           WRITE (5,205) TERM3,FRE,TIM3,FIM,J,I,M,Q,K          !**
            IF (FLJ.NE.1) GO TO 67
          CALL FRAC(KQ,K,J,LRE(J),LIM(J),1,UK,UK1,PVJX1,PVJ1X1,VQK,VQK1,
         1FRE,FIM)
            TERM31=TERM31+FRE
            TIM31=TIM31+FIM
      67    IF (FLK.EQ.1) K=K+1
      68    IF (K.EQ.NS) GO TO 70
            K=K+1
            GO TO 61
      70    CONTINUE
C           WRITE (5,206) TERM1,TERM2,TERM3,TIM1,TIM2,TIM3      !**
C 206       FORMAT (1X,'TERM1,TERM2,TERM3,TIM1,TIM2,TIM3',/,1X,6F15.6) !**
            PZXRE=TERM1+TERM2+TERM3
            PZXIM=TIM1+TIM2+TIM3
            CALL ZK (KQ,M,LRE(M),LIM(M),ZRE,ZIM)
C           WRITE (5,207) SUMQ1,ZRE,PZXRE,ZIM,PZXIM,J,I,M,Q     !**
C 207       FORMAT (1X,'SUMQ1,ZRE,PZXRE,ZIM,PZXIM',/,1X,5F15.6,'J,I,M,Q',4I2)!*'
            SUMQ1=SUMQ1+ZRE*PZXRE+ZIM*PZXIM
C           WRITE (5,208) SUMQ1,J,I,M,Q                         !**
C 208       FORMAT (1X,'SUMQ1=',F15.6,'J,I,M,Q',4I2)            !**
            IF (FLJ.NE.1) GO TO 80
            PZXRE=TERM11+TERM21+TERM31
            PZXIM=TIM11+TIM21+TIM31
            SUMQ2=SUMQ2+ZRE*PZXRE+ZIM*PZXIM
```

```
   80     CONTINUE
          PJ2(I,J)=2*P(M)*SUMQ1+PJ2(I,J)
          IF (FLJ.EQ.1) PJ2(I,JC)=2*P(MC)*SUMQ2+PJ2(I,JC)
C         WRITE (5,203) M,I,J,PJ2(I,J)                  !**
C  203    FORMAT(1X,'==M=',I2,'==I=',I2,'==J=',I2,'PJ2(I,J)=',F15.6) !**
          IF (FLM.EQ.1) M=M+1
          IF (M.EQ.NS) GO TO 110
          M=M+1
          GO TO 40
  110     CONTINUE
          IF (FLJ.EQ.1) J=JC
          IF (J.EQ.NS) GO TO 120
          J=J+1
          GO TO 10
  120     DO 130 II=1,NI
          DO 130 IJ=1,NS
          G(II,IJ)=PJ1(II,IJ)+PJ2(II,IJ)
  130     CONTINUE
C         CALL USWFM (11HMATRIX [G]:,11,G,10,NI,NS,4)     !**
          CALL DBNORM (NI,NS)
          CALL USWFM (16HGradient matrix:,16,G,10,NI,NS,4)  !**
          RETURN
          END
```

```
C************************************************************
C************************************************************
        SUBROUTINE ZK(Q,J,RELJ,XIMJ,ZRE,ZIM)
C-Function: Expression evaluater for Mode 7.
C-IMSL routines called: -
C-Spectral Assignment routines: COMDIV, and Function T.
C-Logical devices; Input Unit:   -    Output Unit:   (5)
C-            Storage Unit(s): -
C-Random Access Files: -
        REAL V(10,10),U(10,10)
        REAL VJ(10),VJ1(10),UK(10),UK1(10),LRE(10),LIM(10)
      REAL XX(10,10),VA(20),E(20),X(20),WJ(10)
      REAL W(10,10),VINV(10,10)
        REAL A(10,10),B(10,10),C(10,10)
        INTEGER Q,FLJ,FLK
        COMMON/SYS/A,B,C,ZERO,IDGT,NS,NI,NO
        COMMON/EIG/LRE,LIM/LEG/U
        COMMON/VEC/VA,E,X,WJ,W,XX,V,VINV
        FLJ=0
        FLK=0
        JC=J+1
        ZRE=FLOAT(0)
        ZIM=FLOAT(0)
        IF (ABS(XIMJ).GT.ABS(ZERO)) FLJ=1
        DO 900 K=1,NS
        IF (FLK.NE.1) GO TO 10
        FLK=0
        GO TO 900
   10   CONTINUE
C       WRITE (5,3) J,K,LRE(K),LIM(K)                 !**
C   3   FORMAT (1X,'----------J=',I2,'K=',I2,'LAMBDA-K',2F15.6) !**
        IF (ABS(LIM(K)).GT.ABS(ZERO)) FLK=1
        IF (K.EQ.J) GO TO 900
        KC=K+1
        DO 100 IV=1,NS
        VJ1(IV)=FLOAT(0)
        VJ(IV)=V(IV,J)
        IF (FLJ.EQ.1) VJ1(IV)=V(IV,JC)
        UK(IV)=U(IV,K)
        IF (FLK.EQ.1) UK1(IV)=U(IV,KC)
  100   CONTINUE
        VQK=V(Q,K)
        VQK1=FLOAT(0)
C       CALL USWFV(11HVECTOR  VJ:,11, VJ,NS,1,4)      !**
C       CALL USWFV(11HVECTOR VJ1:,11,VJ1,NS,1,4)      !**
C       CALL USWFV(11HVECTOR  UK:,11, UK,NS,1,4)      !**
C       CALL USWFV(11HVECTOR UK1:,11,UK1,NS,1,4)      !**
C       WRITE (5,2)  Q,K,VQK,VQK1
C     2 FORMAT (1X,2HQ=,I2,2HK=,I2,4HVQK=,F15.6,5HVQK1=,F15.6)
        IF (FLK.EQ.1) VQK1=V(Q,KC)
        IF (FLJ.NE.1.AND.FLK.NE.1) GO TO 200
        TRR=T(1,VJ,UK)
        TCC=FLOAT(0)
        IF (FLK.EQ.1) TCC=T(1,VJ1,UK1)
        TRC=T(1,VJ1,UK)
        TCR=FLOAT(0)
```

```
            IF (FLK.EQ.1) TCR=T(1,VJ,UK1)
C           WRITE (5,4) K,J,TRR,TCC,TRC,TCR                   !**
C     4     FORMAT (1X,'------------------K=',I2,'J=',I2,'TRR---,TCC---,TRC
C        1---,TCR',/,27X,4F15.6)                                !**
            A1=(TRR-TCC)*VQK-(TRC+TCR)*VQK1
            B1=(TRC+TCR)*VQK+(TRR-TCC)*VQK1
            A2=RELJ-LRE(K)
            B2=LIM(K)-XIMJ
            CALL COMDIV(A1,B1,A2,B2,A3,B3)
            ZRE=ZRE+A3
            ZIM=ZIM+B3
            GO TO 900
      200   ZRE=ZRE+(T(1,VJ,UK)*VQK)/(RELJ-LRE(K))
     900    CONTINUE
C           WRITE (5,1) Q,J,ZRE,ZIM                           !**
C     1     FORMAT (1X,1HZ,I2,I2,6H: ZRE=,F15.6,6H: ZIM=,F15.6)  !**
            RETURN
            END
```

```
C***************************************************************
C***************************************************************
      SUBROUTINE PU(PUMAT)
C-Function: Returns [PUMAT]=p[U]/p[X] .
C-IMSL routines called: VMULFF,UERTST,(USWFV,USWFM).
C-Spectral Assignment routines: TRAN.
C-Logical devices; Input Unit:   -    Output Unit:   (5)
C-           Storage Unit(s): -
C-Random Access Files: -
     REAL XX(10,10),VA(20),E(20),X(20),WJ(10)
     REAL W(10,10),VINV(10,10),V(10,10),VECTOR(10)
      REAL A(10,10),B(10,10),C(10,10)
      REAL AUX1(10,10),AUX2(10,10),AUX3(10,10),PUMAT(10,10)
      COMMON/SYS/A,B,C,ZERO,IDGT,NS,NI,NO
      COMMON/VEC/VA,E,X,WJ,W,XX,V,VINV
      COMMON/AUX/AUX1,AUX2,AUX3
C     WRITE (5,1) J                          !**
C 1   FORMAT (1X,'SUBROUTINE PU, J=',I2)     !**
C     CALL USWFM (10HMATRIX W :,10,W,10,NS,NS,4)       !**
      CALL VMULFF(W,VINV,NS,NS,NS,10,10,AUX1,10,IER)
      CALL UERTST(IER,6HVMULFF)
      CALL VMULFF(VINV,AUX1,NS,NS,NS,10,10,AUX2,10,IER)
      CALL UERTST(IER,6HVMULFF)
      CALL tran(AUX2,NS,NS)
      DO 10 IV=1,NS
      DO 10 JV=1,NS
      PUMAT(IV,JV)=-AUX2(IV,JV)
  10  CONTINUE
C     CALL USWFV(14HVECTOR VECTOR:,14,VECTOR,NS,1,4)      !**
      RETURN
      END
```

```
C**********************************************************************
C**********************************************************************
        SUBROUTINE COMDIV(A1,B1,A2,B2,XRE,XIM)
C-Function: COMPLEX DIVISION, XRE+JXIM=(A1+JB1)/(A2+JB2) .
C-IMSL routines called: UERTST,VMULFM,(USWFM,USWFV).
C-Spectral Assignment routines: -
C-Logical devices; Input Unit:   -    Output Unit:   (5)
C-                Storage Unit(s): -
C-Random Access Files: -
        IF (ABS(B1).GT.FLOAT(0).OR.ABS(B2).GT.FLOAT(0)) GO TO 10
        XIM=FLOAT(0)
        XRE=A1/A2
        GO TO 20
   10   XM=SQRT((A1**2+B1**2)/(A2**2+B2**2))
        XT=ATAN(B1/A1)-ATAN(B2/A2)
        XRE=XM*COS(XT)
        XIM=XM*SIN(XT)
   20   CONTINUE
C        WRITE (5,1) A1,B1,A2,B2,XRE,XIM        !**
C   1   FORMAT (1X,F15.6,2H+J,F15.6,1H/,F15.6,2H+J,F15.6,1H=,/    !**
C     1,20X,F15.6,2H+J,F15.6)                            !**
        RETURN
        END
```

```
C*************************************************************
C***********************************************************
      FUNCTION T(ID,VEC1,VEC2)
C-Function: Evaluates T=<V1,V2> where  V1 and V2 are derminrd by
C          the choice of ID,VEC1, and VEC2.
C-IMSL routines called: UERTST,VMULFM,(USWFV,USWFM).
C-Spectral Assignment routines: -
C-Logical devices; Input Unit:   -    Output Unit:   (5)
C-           Storage Unit(s): -
C-Random Access Files: -
      REAL VEC(10),VEC1(10),VEC2(10),DAD(10,10),TX(1,1),DBD(10,10)
      REAL A(10,10),B(10,10),C(10,10),DAHD(10,10)
      REAL AUX1(10,10),AUX2(10,10),AUX3(10,10),AUX4(10,10)
      COMMON/SEN/DAD,DBD,DAHD/AUX/AUX1,AUX2,AUX3/AAUX/AUX4
      COMMON/SYS/A,B,C,ZERO,IDGT,NS,NI,NO
C     CALL USWFM (11H[dAHAT/dP]:,11,DAHD,10,NS,NS,4)      !**
C     CALL USWFV (12HVECTOR VEC1:,12,VEC1,NS,1,4)     !**
C     CALL USWFV (12HVECTOR VEC2:,12,VEC2,NS,1,4)     !**
      GO TO (1,2,3),ID
    1 CALL VMULFM (DAHD,VEC2,NS,NS,1,10,10,VEC,10,IER)
      GO TO 10
    2 CALL VMULFM(AUX4,VEC2,NS,NS,1,10,10,VEC,10,IER)
      GO TO 10
    3 CALL VMULFM(AUX3,VEC2,NS,NS,1,10,10,VEC,10,IER)
   10 CALL UERTST (IER,6HVMULFM)
C     CALL USWFV (12HVECTOR VEC :,12,VEC ,NS,1,4)     !**
      CALL VMULFM (VEC1,VEC,NS,1,1,10,10,TX,1,IER)
      CALL UERTST (IER,6HVMULFM)
      T=TX(1,1)
C     WRITE (5,11) ID,T                      !**
C 11  FORMAT (1X,'ID=',I2,5X,'T=',F15.6)           !**
      RETURN
      END
```

```
C*********************************************************************
C*********************************************************************
      SUBROUTINE TRAN(A,IM,IN)
C-Function: Returns the Transpose of [A] in itself.
C-IMSL routines called: -
C-Spectral Assignment routines: -
C-Logical devices; Input Unit:   -    Output Unit:   -
C-                Storage Unit(s): -
C-Random Access Files: -
      REAL A(10,10),AT(10,10)
      DO 10 I=1,IM
      DO 10 J=1,IN
      AT(J,I)=A(I,J)
   10 CONTINUE
      DO 20 I=1,IN
      DO 20 J=1,IM
      A(I,J)=AT(I,J)
   20 CONTINUE
      RETURN
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
c****************************************************************
C****************************************************************
      SUBROUTINE SENS
C-Function: Calculates d[AHAT]/dP.
C-IMSL routines called: UERTST,VMULFF,(USWFM).
C-Spectral Assignment routines: -
C-Logical devices; Input Unit:    -    Output Unit:    (5)
C-              Storage Unit(s): -
C-Random Access Files: -
      REAL DAD(10,10),DBD(10,10),DAHD(10,10),F(10,10),AHAT(10,10)
      REAL A(10,10),B(10,10),C(10,10)
      COMMON/SEN/DAD,DBD,DAHD/AUG/F,AHAT
      COMMON/SYS/A,B,C,ZERO,IDGT,NS,NI,NO
      CALL VMULFF (DBD,F,NS,NI,NS,10,10,DAHD,10,IER)
      CALL UERTST (IER,6HVMULFF)
C     CALL USWFM (12H[dB/dP]*[F]:,12,DAHD,10,NS,NS,4)    !**
      DO 25 II=1,NS
      DO 25 IJ=1,NS
      DAHD(II,IJ)=DAD(II,IJ)+DAHD(II,IJ)
   25 CONTINUE
C     CALL USWFM (11H[dAHAT/dP]:,11,DAHD,10,NS,NS,4)      !**
      RETURN
      END
```

ORIGINAL PAGE IS OF POOR QUALITY

```
C*************************************************************
C*************************************************************
      SUBROUTINE FRAC(IQ,IA,IB,RELB,XIMB,ID,UA,UA1,VB,VB1,
     1VQA,VQA1,FRE,FIM)
C-Function: Expression evaluater for MODE 7.
C-IMSL routines called: -
C-Spectral Assignment routines: COMDIV, nad Function T.
C-Logical devices; Input Unit:   -    Output Unit:   (5)
C-               Storage Unit(s): -
C-Random Access Files: -
     REAL XX(10,10),VA(20),E(20),X(20),WJ(10)
     REAL W(10,10),VINV(10,10),V(10,10),LRE(10),LIM(10)
      REAL A(10,10),B(10,10),C(10,10)
      REAL UA(10),UA1(10),VB(10),VB1(10)
      COMMON/SYS/A,B,C,ZERO,IDGT,NS,NI,NO
      COMMON/VEC/VA,E,X,WJ,W,XX,V,VINV/EIG/LRE,LIM
      FRE=FLOAT(0)
      FIM=FLOAT(0)
      IA1=IA+1
C     WRITE (5,1) RELB,XIMB,VQA,VQA1,ID          !**
C   1 FORMAT (1X,'RELB,XIMB,VQA,VQA1',4F15.6,'ID=',I2)    !**
      IF (IA.EQ.IB) GO TO 99
     IF (ABS(XIMB).GT.ABS(ZERO).OR.ABS(LIM(IA)).GT.ABS(ZERO)) GO TO 10
      FRE=(VQA*T(ID,VB,UA))/(RELB-LRE(IA))
C     WRITE (5,2) FRE,IQ,IA,IB                   !**
C   2 FORMAT (1X,'FRE=',F15.6,5X,'IQ,IA,IB',3I2) !**
      GO TO 99
   10 TRR=T(ID,VB,UA)
      TCC=FLOAT(0)
      IF (ABS(XIMB).GT.ABS(ZERO).AND.ABS(LIM(IA)).GT.ABS(ZERO)) TCC=
    1T(ID,VB1,UA1)
      TCR=FLOAT(0)
      IF (ABS(XIMB).GT.ABS(ZERO)) TCR=T(ID,VB1,UA)
      TRC=FLOAT(0)
      IF (ABS(LIM(IA)).GT.ABS(ZERO)) TRC=T(ID,VB,UA1)
      A1=(TRR-TCC)*VQA-(TCR+TRC)*VQA1
      B1=(TCR+TRC)*VQA+(TRR-TCC)*VQA1
      A2=RELB-LRE(IA)
      B2=LIM(IA)-XIMB
      CALL COMDIV (A1,B1,A2,B2,FRE,FIM)
   99 RETURN
      END
```